CLOUD WEB SERVER FOR IOT SMART LABORATORY SYSTEM

LEE YI MING

Bachelor of Electronics Engineering Technology (Computer System) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THE	SIS AND COPYRIGHT		
Author's Full Name :	LEE YI MING		
Date of Birth :			
Title :	SMART LABORATORY SYSTEM		
Academic Session :	SENIOR DESIGN PROJECT		
I declare that this thesis is cla	assified as:		
⊠ CONFIDENTIAL	(Contains confidential information under the Official Secret		
⊠ RESTRICTED	(Contains restricted information as specified by the		
\square OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)		
I acknowledge that Universit	i Malaysia Pahang reserves the following rights:		
 The Thesis is the Property of Universiti Malaysia Pahang The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only. The Library has the right to make copies of the thesis for academic exchange. 			
Certified by:			
	AMRAN BIN ABDUL HADI		
(Student's Signature)	(Supervisor's Signature)		
	AMRAN BIN ABDUL HADI		
New IC/Passport Number Date: 15 January 2022	er Name of Supervisor Date: 25 February 2022		

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the Bachelor of Electronics Engineering Technology (Computer System) With Honors.

AMRAN BIN ABDUL HADI

(Supervisor's Signature) Full Name : MR. AMRAN BIN ABDUL HADI Position : LECTURER Date : 25 February 2022



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature) Full Name : LEE YI MING ID Number : TG18040 Date : 25 JANUARY 2022

CLOUD WEB SERVER FOR IOT SMART LABORATORY SYSTEM

LEE YI MING

Thesis submitted in fulfillment of the requirements for the award of the degree of Bachelor of Electronics Engineering Technology (Computer System) With Honours

Faculty of Electrical & Electronics Engineering Technology

UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2022

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my supervisor, Mr. Amran Bin Abdul Hadi, who is very helpful in allowing me to express my engineering skills in this senior design project (SDP). His unlimited patience, support, and motivation were vital when I encountered difficulties throughout completing this project. It is an honor for me to work under his supervision which shaped the present work as its show.

My appreciation also goes to my project members, Carlos Enseriban and Muhammad Aqil Bin Muhammad Suhud. They gave encouragement and great teamwork to improve the project from the beginning until the end to achieve the best result.

Besides, special dedication to FTKEE UMP for allowing me to be involved in a project that helps me see better in the engineering world. Finally, I hope all the experience and knowledge gained through this project will give me the utmost benefit to my future career.

Last but not least, I would like to express my utmost gratitude to my family members for all the support spiritually and financially throughout my studies, especially for the completion of the senior design project.

ABSTRAK

Teknologi IoT (Internet of Things) sedang mengalami pertumbuhan yang luar biasa apabila ketersambungan berkembang ke seluruh dunia, dan terus melanggar sempadan untuk menunjukkan perkara yang boleh dilakukan oleh teknologi canggih. Kebanyakan makmal kurang menggunakan teknologi IoT dan masih membayar kos yang tinggi untuk memastikan keselamatan. Dalam senario yang sangat biasa, peralatan makmal seperti lampu dan penghawa dingin dibiarkan menyala, walaupun tidak digunakan. Tindakan ini telah menyebabkan peningkatan dalam penggunaan kuasa dan pembaziran tenaga di makmal. Oleh itu, projek yang dicadangkan ini bertujuan untuk membangunkan sistem makmal pintar berasaskan IoT yang menjimatkan kos dan boleh mengurangkan penggunaan kuasa, keselamatan yang diperkukuh, dan memberikan kemudahan kepada pengguna untuk mengawal peralatan makmal walaupun pada jarak yang jauh. Protokol MQTT ialah komunikasi antara pelayan dan klien untuk menerbitkan dan melanggan data. Sistem ini menggunakan NodeMCU untuk menyambungkan beberapa penderia berbeza dan memindahkan data mereka ke pelayan tempatan Raspberry Pi menggunakan Wi-Fi, yang kemudiannya akan dihantar ke pelayan awan. Pengguna boleh terus berhubung dengan makmal dari semasa ke semasa dengan mengakses laman web antara muka mesra pengguna melalui telefon pintar atau komputer riba. Data masa nyata yang dikumpul akan dipaparkan pada papan pemuka yang dibangunkan dalam Node-RED. Sistem makmal pintar yang dicadangkan berhasrat untuk mengawal peralatan menggunakan Internet dengan cekap dengan kemudahan menyemak data dari tapak web pada bila-bila masa, di mana-mana sahaja.

ABSTRACT

The IoT (Internet of Things) technology is experiencing tremendous growth as the connectivity expands across the world, and it keeps on breaking the boundaries on showing what advanced technology can do. Most of the laboratories lack IoT technology and still pay a high cost to ensure security. In the very common scenario, the appliances of the laboratory such as lights and air conditioners are kept left on, even when not in use. It leads to a rise in power consumption and wastage of energy in the laboratory. Therefore, this proposed project aims to develop a cost-effective and reliable IoT-based smart laboratory system that reduces power consumption, strengthened security, and provides convenience for users to control laboratory appliances even at a distance. MQTT protocol is the communication between the server and client to publish and subscribe to the data. This system utilizes NodeMCU to connect several different sensors and transfer their data to the Raspberry Pi local server using Wi-Fi protocol. It will then send it to cloud server. It enables users to stay connected with the laboratory from time to time by accessing the user-friendly interface website through smartphones or laptops. The realtime collected data will be displayed on the dashboard developed in Node-RED. The proposed smart laboratory system intends to efficiently control appliances using the Internet with the convenience of checking data from the website anytime, anywhere.

TABLE OF CONTENT

DEC	LARATION	
TITI	LE PAGE	
ACK	KNOWLEDGEMENTS	ii
ABS	TRAK	iii
ABS	TRACT	iv
TAB	LE OF CONTENT	v
LIST	Γ OF TABLES	viii
LIST	Γ OF FIGURES	ix
LIST	Γ OF ABBREVIATIONS	xii
LIST	Γ OF APPENDICES	xiii
1	INTRODUCTION	14
1.1	Research Background	14
1.2	Problem Statement	16
1.3	Objectives	16
1.4	Research Scope	17
2	LITERATURE REVIEW	18
2.1	Internet of Things (IoT)	18
2.2	History of Smart Home System	18
2.3	Smart Laboratory System	19
2.4	Related Work	20
	2.4.1 Bluetooth Based Home Automation System	20
	2.4.2 Zigbee Network Multi-Application Sensor in Smart Homes	20

2.4.3 IoT-Based Automation System for Smart Home 21

3	METH	IODOLOGY	22
3.1	System	Description	22
3.2	Selecti	on of Components	23
	3.2.1	Raspberry Pi	23
	3.2.2	ESP32	24
	3.2.3	ESP32-CAM	24
	3.2.4	DHT11 Temperature and Humidity Sensor	25
	3.2.5	PIR Motion Sensor	25
	3.2.6	Relay Module	26
	3.2.7	Light Bulbs	26
3.3	System	n Development	28
	3.3.1	Design and Layout of Prototype	28
	3.3.2	Monitoring System of Prototype	29
3.4	Softwa	are Description	32
	3.4.1	Node-RED	32
	3.4.2	Arduino IDE	32
	3.4.3	Visual Studio	33
	3.4.4	DigitalOcean	34
	3.4.5	FileZilla	34
	3.4.6	Twilio	35
3.5	System	n Architecture	36
3.6	System	n Flowchart	38
3.7	Produc	et Testing	40
3.8	Cost E	stimation	45

RESULTS AND DISCUSSION

4.1	DigitalOcean Cloud Server Setup	46
4.2	Apache and FileZilla Installation	50
4.3	FileZilla File Transfer	53
4.4	Development of Web Page	55
	4.4.1 PHP	55
	4.4.2 CSS	62
	4.4.3 JavaScript	73
4.5	Project Outcome	75
4.6	Discussion	83
5	CONCLUSION	84
5.1	Conclusion	84
5.2	Limitation	85
5.3	Recommendation	85
REFE	RENCES	87

APPENDICES

89

LIST OF TABLES

Table 1: List of Components	23
Table 2: Cost Analysis	45

LIST OF FIGURES

Figure 3.1: Smart Laboratory System	22
Figure 3.2: Raspberry Pi	23
Figure 3.3: ESP32	24
Figure 3.4: ESP32-CAM	24
Figure 3.5: DHT11 Temperature and Humidity Sensor	25
Figure 3.6: PIR Motion Sensor	26
Figure 3.7: Relay Module	26
Figure 3.8: LED Light Bulb	27
Figure 3.9: Smart Laboratory Layout	28
Figure 3.10: DHT11 Temperature and Humidity Sensor Hardware	29
Figure 3.11: ESP32-CAM Hardware	29
Figure 3.12: PIR Motion Sensor Hardware	30
Figure 3.13: Relay Module Hardware	30
Figure 3.14: Light Bulbs Hardware	30
Figure 3.15: Prototype of Smart Laboratory System	31
Figure 3.16: Node-RED programming for DHT11 Sensor	32
Figure 3.17: Arduino Coding of DHT11 Sensor	33
Figure 3.18: Visual Studio Code	33
Figure 3.19: Digital Ocean	34
Figure 3.20: FileZilla	35
Figure 3.21: Twilio	35
Figure 3.22: System Architecture	37
Figure 3.23: System Flowchart	38
Figure 3.24: Testing of DHT11 Sensor with Output Readings	40
Figure 3.25: Testing of ESP-CAM connected with Power Bank	41
Figure 3.26: Testing of ESP-CAM Display	41
Figure 3.27: Circuit connection of PIR motion sensor	42
Figure 3.28: Testing PIR Motion Sensor in Short Distance	42
Figure 3.29: Testing of PIR Motion Sensor in Long Distance	43
Figure 3.30: Circuit Connection of Relay Modules and Light Bulbs	43
Figure 3.31: Testing for Controlling Light Bulbs	44
Figure 4.1: DigitalOcean Sign Up	46
Figure 4.2: Create New Project at DigitalOcean	47

Figure 4.3: Image and Plan	47
Figure 4.4: CPU Options	48
Figure 4.5: Datacenter Region	48
Figure 4.6: Authentication	49
Figure 4.7: Complete Creating Droplet	49
Figure 4.8: Log In as Root	50
Figure 4.9: Default Ubuntu Apache Web Page	51
Figure 4.10: FileZilla Overview	54
Figure 4.11: PHP <head></head>	55
Figure 4.12: PHP <body></body>	56
Figure 4.13: PHP Login Form	57
Figure 4.14: PHP Home Section	57
Figure 4.15: PHP Services Section	58
Figure 4.16: PHP Gallery Section 1	59
Figure 4.17: PHP Gallery Section 2	59
Figure 4.18: PHP About Us Section 1	60
Figure 4.19: PHP About Us Section 2	60
Figure 4.20: PHP Footer	61
Figure 4.21: CSS Overall Section 1	62
Figure 4.22: CSS Overall Section 2	63
Figure 4.23: CSS Header Navigation Bar	64
Figure 4.24: CSS Header Search Bar	64
Figure 4.25: CSS Header Login Form Section 1	65
Figure 4.26: CSS Header Login Form Section 2	65
Figure 4.27: CSS Header Login Form Section 3	66
Figure 4.28: CSS Header Login Form Section 4	66
Figure 4.29: CSS Home Section 1	67
Figure 4.30: CSS Home Section 2	67
Figure 4.31: CSS Home Section 3	67
Figure 4.32: CSS Services Section 1	68
Figure 4.33: CSS Services Section 2	68
Figure 4.34: CSS Gallery Section 1	69
Figure 4.35: CSS Gallery Section 2	69
Figure 4.36: CSS About Us Section	70
Figure 4.37: CSS Footer Section 1	71

Figure 4.38: CSS Footer Section 2	71
Figure 4.39: CSS Media Queries Section 1	72
Figure 4.40: CSS Media Queries Section 2	72
Figure 4.41: CSS Media Queries Section 3	72
Figure 4.42: JavaScript Section 1	73
Figure 4.43: JavaScript Section 2	74
Figure 4.44: Smart Laboratory System Home Web Page	75
Figure 4.45: Search Bar Outcome	75
Figure 4.46: Login Form Outcome	76
Figure 4.47: Web Page Part Services	76
Figure 4.48: Web Page Part Gallery	77
Figure 4.49: Web Page Part About Us	77
Figure 4.50: Web Page Footer	77
Figure 4.51: Side navigation bar	78
Figure 4.52: Temperature and Humidity in Area 1 of Laboratory	79
Figure 4.53: Temperature and Humidity in Area 2 of Laboratory	79
Figure 4.54: ESP32-Camera Display	80
Figure 4.55: No Motion Deteced	80
Figure 4.56: Motion Detected	80
Figure 4.57: Message Alert from Twilio	81
Figure 4.58: Lamp Switches	81
Figure 4.59: Lamp Turned On	82

LIST OF ABBREVIATIONS

IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
NODEMCU	Node-Microcontroller Unit
PIR	Passive Infrared
ESP	Extra-sensory Perception
IDE	Integrated Development Environment
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	PHP Hypertext Preprocessor
SMS	Short Message Service
PLC	Programmable Logic Controller
PC	Personal Computer
НАР	Home Automation Protocol
FTP	File Transfer Protocol
SFTP	Secure File Transfer Protocol

LIST OF APPENDICES

Appendix A:	Gantt Chart for Senior Project	90
Appendix B:	PHP Coding	90
Appendix C	: CSS Coding	95
Appendix D	: JavaScript Coding	105

INTRODUCTION

1.1 Research Background

The invention of the Smart Laboratory System is to upgrade the standard by applying the advanced technology available. It brings many advantages for people in this era that always asked for fast and convenient, at the same time providing comfort and security. The existing smart systems had maximized their functionality to satisfy all human needs to emphasize the industry 4.0 revolution. The acceptance of using automation technology eventually stimulates worldwide economic growth as smart automation systems are getting more competitive. With the implementation of automation, the efficiency and reliability increased with less human intervention needed to do tasks.

The security of the laboratory should be strengthened with the advanced technology available, which is IoT technology. A camera and motion sensor are suggested to use in this system to ensure the security of users. Besides, wastage of energy has become the biggest problem as people tend to forget to switch off the appliances after leaving from one place. People nowadays are living a busy life, and it is very time-consuming to switch off all the appliances as the sockets are normally placed separately. The fact that most of the people do not turn off them when they leave causes wastage of energy. Real-time data monitoring and control using IoT are one of the ways to reduce energy wastage. To overcome this problem, a system with automation control and a monitoring system should be designed and developed so that people can control and monitor the appliances wirelessly.

A spike in electricity usage can be seen in the monthly electricity bill. The monthly electricity bill comes from cumulative daily consumption (Longe et al. 2015). In the very common scenario, the appliances of the laboratory such as lights and air

1

conditioners are kept left on, even when not in use. Imagine as time passes, it eventually leads to a rise in power consumption and wastage of energy in the laboratory. With so much help from the technology, it is only able to solve parts of the problem. If the root cause of the problem of high consumption of energy is not addressed, the problem will still exist, and it will get worse and worse over time as people do not put much attention to its consequences and choose to ignore it. Therefore, the action of spreading public awareness on energy savings (Sanduleac et al. 2017) is important. People should acknowledge energy efficiency, which refers to using less energy to produce the same amount of services or useful output (Patterson 1996). People are encouraged to start by changing their bad habits and behavior, such as always turning off the appliances whenever not in use. Other than that, the use of IoT is the best way to overcome this issue.

This study presents an IoT-based smart laboratory system with the design, fabrication, and validation. It enables IoT automation and monitoring laboratory by using Node-MCU as a microcontroller and Wi-Fi gateway. Several sensors are used to control and monitor various environmental parameters such as temperature and humidity. An ESP32-camera and a PIR motion sensor are attached to the system to enhance laboratory security. A few relays are used to control the activities of laboratory appliances, such as lights. The MQTT protocol acts as the communication between the server and Node-MCU to publish and subscribe the data. Furthermore, a user-friendly interface website is developed to bring convenience in the aspect of the interaction between the users and the smart laboratory as the real-time collected data will be displayed on the dashboard developed in Node-RED. Laboratory users are allowed to view and control the appliance through the website. A simple prototype is constructed to implement and validate the effectiveness of monitoring the appliances of the proposed smart laboratory system. The developed system aims to develop a costeffective and reliable IoT-based smart laboratory system to automate laboratory appliances, strengthen security, and upgrade life quality with technologies.

To summarize, this project is focusing on developing a Smart IoT Laboratory System and monitoring system controlled by Raspberry Pi through the Internet of Things (IoT). By having this smart IoT laboratory system, it provides high-efficiency as the system can be used wirelessly through the internet instead of wired connection. It also provides more safety and security as it can notify the user through SMS or email when there is motion detected. The smart laboratory system will be able to solve the problems in a high efficiency, low cost, and user-friendly way.

1.2 Problem Statement

People nowadays are living a busy life, it is very time-consuming to switch off all the appliances as the sockets are normally placed separately. The fact that the behavior of most people do not turn off the appliances when they are not in use or when they leave causes the problem of wastage of energy. Besides, most people are lack using IoT technology and still paying a high cost for security systems installation. The security should be strengthened with the advanced technology available, which is IoT technology. Difficulties of real-time data monitoring and controlling the systems is one of the problems in this advanced technology world.

1.3 Objectives

- 1. To design a cost-effective and convenience convenient IoT-based smart laboratory system raspberry pi.
- 2. To strengthen the laboratory's security by applying various types of sensors.
- 3. Real-time data monitoring and control using IoT to reduce energy wastage in a laboratory.

1.4 Research Scope

This study presents a cost-effective and reliable IoT-based smart laboratory system with strengthened security, energy savings, and real-time monitoring functions to overcome these problems. It enables IoT automation and monitoring laboratory by using an ESP32 microcontroller with integrated Wi-Fi. Several sensors are used to control and monitor various environmental parameters such as temperature and humidity. The Message Queuing Telemetry Transport (MQTT) protocol act as the communication between the server and client to publish and subscribe the data. Furthermore, a user-friendly interface website is developed to bring convenience in the aspect of the interaction between the users and the smart laboratory as the real-time collected data will be displayed on the dashboard developed in Node-RED. Laboratory users are allowed to view and control the appliances wirelessly through the website. A simple prototype is constructed to implement and validate the effectiveness of monitoring the appliances of the proposed smart laboratory system. The developed system aims to develop a cost-effective and reliable IoT-based smart laboratory system to automate laboratory appliances, strengthen security, and real-time monitoring with technologies.

LITERATURE REVIEW

2.1 Internet of Things (IoT)

Today's high popularity of the Internet of Things (IoT) is using the Internet to the extreme and brings such a big impact on changing the world to better and smarter. It has become an important feature as it is beneficial in many aspects and many different types of smart automation systems have been developed to catch up with the latest trend. With an internet connection, all the tasks became easier and more convenient. IoT technology enables easy access between humans and things (D. Pavithra & Balakrishnan, 2015) for environmental monitoring, security and safety enhancing, wastage managing, as well as real-time monitoring.

2.2 History of Smart Home System

Echo IV, the first smart automation system, was developed in 1966. This device allowed consumers to create computing shopping lists, control home temperature and turn appliances on and off. The kitchen computer that was created in 1969 could create recipes. In 1991, "Gerontechnology" combined gerontology with technology made the lives of senior citizens easier. In the early 2000s, smart home technology began to increase the popularity and different technologies emerged and were slowly integrated into homes. Smart homes started to become affordable options and therefore viable technologies for many consumers.

Security and efficiency are the main reasons behind the increase in smart home technology use. According to Statista, the future of smart homes is exciting – there will be around 31 billion devices connected to the internet by 2020, and that number is supposed to grow to 75.4 billion by 2025. It is not surprising that the smart home system

2

has such high demand in the future because technology makes things easier and more convenient.

2.3 Smart Laboratory System

Since the concept of an IoT-based system is almost the same, it can be developed in different ways and different places such as offices and laboratories.

(Poongothai and Subramanian 2018) delivered IoT-based Smart Laboratory with the use of ESP8266, Arduino UNO, relays, current transformers, Raspberry Pi 3, and sensors. A dashboard is developed in Node-RED or ANDROID STUDIO mobile application to enable the users to control and monitor the devices of the laboratory. Node MCU is also coded to monitor and update the temperature, humidity, and light intensity inside the laboratory. The appliances in the laboratory can be remotely monitored and controlled, thereby reducing their energy consumption considerably.

(Amruta N. Banagar and Rajshankar Khattar 2020) developed an IoT-based Smart Laboratory System. The system is built to emphasize the environmental parameters such as temperature and light intensity of the laboratory and promotes effective power consumption. The Raspberry pi 3B is used as the microcontroller with connected sensors including the PIR motion sensor, TSL2561 luminosity sensor, and DHT11 temperature sensor.

2.4 Related Work

2.4.1 Bluetooth Based Home Automation System

Bluetooth wireless technology is expected to change the way people think about digital gadgets in their homes and offices (Sriskanthan, Tan, and Karande 2002). This study used Bluetooth links as the communication between the host and the microcontroller. The author developed the Home Automation Protocol (HAP) to facilitate master-slave communication in the automation networks. Multiple appliances are connected to a single Bluetooth, with one or more microcontrollers monitoring them. This system allows the user to monitor and control several appliances with a Bluetooth connection.

There are many smart devices are used in the present laboratories such as smart boards, smart projectors, smart tablets, etc. (M. Elatawy 2020) proposed an IoT-based laboratory system that aims to control the general activities of the laboratory. The Arduino UNO, a Bluetooth connection, sensors, and bulbs are used in developing this system. All components are installed and connected to the smartphone Bluetooth. The results indicate that with Bluetooth connection, laboratory activities can be controlled and monitored.

2.4.2 Zigbee Network Multi-Application Sensor in Smart Homes

(Jhang et al. 2017) proposed a low-cost Zigbee network-based to detect the home water leakage and opening/closing of the door. The Zigbee is described as a good characteristic of a mesh network architecture and it provides better performance in transmission rate compared to others but the transmission range of other wireless communications such as Wi-Fi and Bluetooth are way better than Zigbee. This study proved that the electric current consumption and firmware code density of the proposed sensor was better than the other third-party sensor.

2.4.3 IoT-Based Automation System for Smart Home

(Jabbar et al. 2018) presented a low-cost Wi-Fi-based automation system for smart homes that is capable of remotely monitoring and controlling home appliances using an Android-based application, Virtuino. Arduino Mega microcontroller along with WI-FI module ESP8266 is focused in this study for controlling the home appliances. The user-friendly interface Virtuino application works efficiently with Arduino Mega to control and monitor the electrical appliances such as the bulb and fan using a smartphone. Arduino controller is programmed to interact with the user-friendly interface Virtuino application.

METHODOLOGY

3

3.1 System Description

Figure 3.1 shows the proposed smart laboratory system is categorized into 3 main key elements which included security system, energy savings, and real-time monitoring.

For the security system, the ESP32 Camera plays the role of recording and displaying the real-time vision of the laboratory. The security can be strengthened by using a PIR motion sensor to detect the motion and an SMS alert will be sent to the authorities when there is any motion detected.

For the energy saving category, it is convenient for the laboratory user to control and monitor the appliances even at a distance. Eventually, the energy consumption can be reduced. It solves the problem as people tend to forget to switch off the laboratory appliances even not in use.

Real-time monitoring brings convenience as people can control and monitor everything through a wireless connection. Laboratory users can access the website anytime as all the laboratory data such as temperature readings, camera display, motion detection, and switch are displayed on the Node-RED dashboard.



Figure 3.1: Smart Laboratory System

3.2 Selection of Components

In the early stages of prototype device design, component selection is critical. Since each of the components serves a distinct purpose, it is best to select the most suitable and appropriate component based on the project criteria. In this project, the selected components and quantity needed are listed down in Table 1.

ruble i. List of components	Table	1:1	List	of	Com	ponents
-----------------------------	-------	-----	------	----	-----	---------

No.	Components	Quantity
1.	Raspberry Pi	1
2.	ESP32	4
3.	ESP32-CAM	1
4.	DHT11 Temperature and Humidity Sensor	2
5.	PIR Motion Sensor	1
6.	Relay Module	2
7.	LED Light Bulb	2

3.2.1 Raspberry Pi

A single-board computer Raspberry Pi is often used for projects that are related to smart automation projects. It is a low-cost tiny computer that is capable of doing work and functions like a desktop computer that can control appliances (Ruwaida and Minkkinen 2013). Before everything starts, the LAMP stack server, a software bundle is needed to be installed on the Raspberry Pi that is used for web development. LAMP are Linux, Apache, MySQL, and PHP. Linux is the operating system, Apache is the HTTP web server, MySQL is the database management, and PHP is the programming language.



Figure 3.2: Raspberry Pi

3.2.2 ESP32

A microcontroller is needed in this project. A few units of ESP32 microcontroller are selected to use in this project because it has integrated Wi-Fi and supports Bluetooth connectivity. It functions to perform for the data collection of the device with a sensor attached to it. Next, it will be then programmed by using the Arduino IDE Software.



Figure 3.3: ESP32

3.2.3 ESP32-CAM

The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and micro SD card socket. It is inexpensive and easy to use, perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition. The ESP32-CAM module has fewer I/O pins than the previous ESP-32 module we looked at. Many of the GPIO pins are used internally for the camera and the micro SD card port. Another thing missing from the ESP32-CAM module is a USB port.



Figure 3.4: ESP32-CAM

3.2.4 DHT11 Temperature and Humidity Sensor

A DHT11 temperature and humidity sensor is selected as it is the best option due to the advantages such as low cost, fast response, high accuracy, and precise calibration. This temperature sensor is used to detect the temperature and humidity of the surroundings of the laboratory.



Figure 3.5: DHT11 Temperature and Humidity Sensor

3.2.5 PIR Motion Sensor

A sensor that can detect motion in a certain range is required in this project for security purposes. A Passive Infrared (PIR) motion sensor is selected to detect the motion in this project. PIR sensors are commonly used to detect human or object movement in or out of the sensors range. The detection range is of the sensor is up to 6 meters. PIR motion sensor meets the requirement as they are small, inexpensive and low-power. In addition, the adjustment of sensitivity and delay time of the PIR motion sensor can be made at the yellow trim pot of the motion sensor. For time delay adjustment, turning the left trim pot clockwise can increase the delay time while turning counter-clockwise to decrease the delay time. Besides, the sensitivity of the motion sensor can be adjusted by turning the right yellow trim pot to the direction to clockwise will make it more sensitive while turning it to counter-clockwise direction will reduce the sensitivity.



Figure 3.6: PIR Motion Sensor

3.2.6 Relay Module

A few units of 1-channel 5V DC relay module are needed in this project to perform the switching of the actuators, which are light bulbs. The relay is energized or reenergized based on the received signals from the NodeMCU, which receives the commands from the user or the sensors. The relay board overcomes the limitation of the control voltage generated by the controller.



Figure 3.7: Relay Module

3.2.7 Light Bulbs

LED light bulbs are needed in this project to act as the lights in the smart laboratory. LED light bulb is selected to be used in this project because it is highly energy-efficient as it used less electricity in giving the same light output compared to the normal light bulbs. It is low cost at the same time produced less heat but more light. In this study, the LED light bulbs are used as replacements for actuators to read the output.



Figure 3.8: LED Light Bulb

3.3 System Development

The development of the smart laboratory system includes the detailed design layout, implementation of monitoring systems in the smart laboratory system prototype, and lastly product testing based on the components selection so that the design objectives in this study can be achieved.

3.3.1 Design and Layout of Prototype

A layout of the smart laboratory is simply sketched as shown in Figure 3.9. From the layout plan, the smart laboratory consists of two areas. The front part of the laboratory is named area 1, while the back of the laboratory is named area 2. The reason for separating two areas for the laboratory is to test the DHT11 temperature and humidity sensors. They should be giving two different reading outputs if the temperature and humidity of the 2 laboratory areas are different. A PIR motion sensor is placed near the laboratory to display a real-time full eye-view vision of the laboratory. The lightbulbs are also placed in front of the laboratory.



Figure 3.9: Smart Laboratory Layout

3.3.2 Monitoring System of Prototype

There are several monitoring systems in this project which included the DHT11 temperature and humidity sensor, ESP32, ESP32-CAM, PIR motion sensor, relay module, and light bulbs are all shown from Figure 3.10 to Figure 3.14. A process of PCB soldering, which is melting some solder alloy on top of the connection is done to make sure the circuit is more stable and reliable. As the soldering iron melts the metal, it will then join the components together.



Figure 3.10: DHT11 Temperature and Humidity Sensor Hardware



Figure 3.11: ESP32-CAM Hardware



Figure 3.12: PIR Motion Sensor Hardware



Figure 3.13: Relay Module Hardware



Figure 3.14: Light Bulbs Hardware

The plywood is chosen to use for the prototype because it is easy to get at an affordable price. Easy to cut is another reason for choosing plywood. It eventually saves a lot of time and energy from spending time to cut the plywood. The prototype is also pasted with wrapping paper to cover the rough surface of plywood to improve the prototype appearance. The prototype of the smart laboratory system is shown in Figure 3.15.

All of the monitoring systems as previously stated such as sensors and actuators are all well prepared but they are not installed in the specific places of the prototype. It is better to place them with freedom so that the output differences can be easily observed. All the monitoring systems will perform their respective functions so that they can successfully achieve the objectives of this project. The wire connection between all the components in the prototype is installed and ready to be tested.



Figure 3.15: Prototype of Smart Laboratory System

3.4 Software Description

3.4.1 Node-RED

Node-RED is an open-source block programming software that can create various flow-based functions by wiring together the hardware devices, especially the most popular IoT devices like sensors, cameras, and wireless routers (Ferencz and József 2020). Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Besides, Node-RED Dashboard provides a user-friendly interface that allows users to create a simple interface without the knowledge of Internet Programming such as HTLM and CSS. Figure 3.16 shows the Node-RED programming flow for the DHT11 temperature and humidity sensor. The readings of data will be displayed at the right side of the Node-RED system flow.



Figure 3.16: Node-RED programming for DHT11 Sensor

3.4.2 Arduino IDE

Arduino IDE (Integrated Development Environment) is open-source software that is used to write codes, compile codes, and upload them to the NodeMCU board. In this project, Arduino IDE is used for coding, debugging, and testing the functionalities of a smart laboratory system with the components used. To communicate with DHT11 sensors, the DHT Sensor Library needed to be installed. Figure 3.17 shows a part of the Arduino coding of the DHT11 temperature and humidity sensor.


Figure 3.17: Arduino Coding of DHT11 Sensor

3.4.3 Visual Studio

Visual Studio Code is a free coding editor software that can be used to write coding in any programming language using only one editor. It is convenient as there is no need to switch to other editors just to use different programming languages. There are many languages supported by the Visual Studio Code, including HTML, CSS, JavaScript, PHP, Python, and more. In this project, there are three languages used such as PHP, CSS, and JavaScript as shown in Figure 3.18. PHP code is executed on the server, and the result is returned to the browser as plain HTML. CSS is a style sheet language that is used to style and make the web page looks more presentable. JavaScript is a set of statements that performs a task.

×1	File Edit Selection View Go Run Terminal	Help index.php - Webpage - Visual Studio Code	- a x
G		🖤 index.php 🗙 🕴 newcss.css 🛛 🥵 newjavascript.js	
-	✓ WEBPAGE	new try > 🐨 index.php	
0	✓ new try	1 <idoctype html=""></idoctype>	
\sim	Et amranina	2 <html lang="en"></html>	
	E salifi		
ુરુ	e aqujii		
	⇒ canos,m	5 <meta content="width=device-width, initial-scale=1.0" name="viewport"/>	
1	🖷 index.php	6 <title>UMP Smart Laboratory</title>	Baggrann-
×.	Iogo.png		JARDShare.
00	# newcss.css	8 <link href="<u>https://unpkg.com/swiper/swiper-bundle.min.css</u>" rel="stylesheet"/>	Bellevil susceptor
	newindex.html		
	JS newjavascript.js	10 (1 font avesome cdn link>	
	🖾 p1.jpeg	11 Clink rel= stylesneet inret= <u>nttps://conjs.cloudriare.com/ajax/libs/tont-awesome/5.15.3/css/all.min.cc</u>	<u>22</u> 250777
	🖬 p2.jpeg		
	D p3.jpeg	14 (link rel="stylesheet" hefe="newcss.css")	
	🖘 p4.jpeg		
	video1.mp4		Section
	video2 mo4		
	n nine and not		
	o video t mot		
	• video4.mp4		
	Video5.mp4		Jane Contract
	*2 waheb.jpg		aprox.
	l≞ yimingijiit	23 (UIV IOE MENU-DAIN CLASSE TAS TA-DAINS X/(UIV)	
	 index.html 	27 // heaf="#"\/ing enc="logo_pag" class="logo"\//a\	
	🔄 labbg.jpg		
	🖼 logo.png	27 <nay class="navbar"></nay>	
	smartlab.html	28 Home	
0	🖤 smartlab.php	29 Services	
V 1	# style.css	30 Gallery	
000		31 About Us	
563			
	A 0		cour out A O

Figure 3.18: Visual Studio Code

3.4.4 DigitalOcean

DigitalOcean is a cloud hosting platform that provides developers the cloud services. The advantage of DigitalOcean is that it has unlimited domains as users can host multiple websites in a single droplet. In this study, the Ubuntu server is chosen as this project's cloud server. By creating a droplet, a new IP address for the cloud server was created, which is 178.128.50.80 as shown in Figure 3.19.

ධ	Q Search by resource name or public IP (CtrI+B) Create ~	Image: My Team My Team Credit: \$84.55 (expires in 27 days) Image: My Team
PROJECTS ^ Smart Laborato + New Project	Smart Laboratory System DEFAULT Class project / Educational purposes / Update your project information under Settings	ightarrow Move Resources
	Resources Activity Settings	
Settings	DROPLETS (1)	
Billing API	• 👌 smartlab 178.128.50.80	۵
Marketplace 7	Create something new	Learn more
	Create a Managed Database Worry-free database management Deliver data with scalable object storag	Product Docs Technical overviews, how-tos, release notes, and support material
	Spin up a Load Balancer Distribute traffic between multiple Droplets	Tutorials DevOps and development guidelines

Figure 3.19: Digital Ocean

3.4.5 FileZilla

FileZilla is open-source software that is used for files transferring to or from a computer by using File Transfer Protocol (FTP) or Secure File Transfer Protocol (SFTP). It is very time-consuming to transfer a large number of files one by one. Instead, FileZilla is used in this study to transfer all the files over the web by just dragging all the files that needed to be transferred as shown in Figure 3.20.

5 sftp://root@	0178.128.50.80 - FileZilla											-	٥	×
File Edit Vie	ew Transfer Server Book	marks Help												
111 - R) 🐛 🔱 🔳 🔍	o 🔥											
Host: sftp://17	78.128.50.80 Username: ro	ot Pass	word:	•• Port:	Quickconne	ct 💌								
Status: Ret	trieving directory listing of "/"	var"												^
Status: Dir	ectory listing of "/var" succes	sful												
Status: Ret	trieving directory listing of "/	var/www"												
Status: List	ting directory /var/www													
Status: Dir Status: Ret	rectory listing of "/var/www" trieving directory listing of "/	successful var/www/html*												
Status: List	ting directory /var/www/htm	d												
Status: Dir	rectory listing of "/var/www/ł	ntml" successful												
1 1 2 0		10 1000000000				D 1 1								•
Local site: C:\	Users\user\Desktop\UIVI\Yea	r4 Sem I\SUP2\Webpag	e\new try\		×	Remote site:	/var/www/ntm	าเ						~
	lect	inical Report			^	~ ?	log							^
	ine:	iis				1	mail							
		age to a				2	opt							
	Softwar	e Engineering			~	2	snan							
Filename	Filesi	re Filetyne	Last modified		^	?	spool							
	1103	ie metype	Last mounted			?	tmp							- 1
	54.22		01/16/22 14:11:40		_	ģ 📃	www							
amian.jpg	34,51	I JPG File	01/16/22 14:11:49		_		- 📊 html							~
aquigm	32,44	1 IEIE Eile	01/16/22 14:12:12		_	Filename	* 1	Filesize	Filebyne	Last modified	Permissions	Owner/Group		^
index php	7.4	1 DHD Source File	01/26/22 09:47:17		_			Thesize	racype	Lust mounicu	1 chillippions	office, oroup		
	123.2	B PNG File	06/01/21 12:14:44					6 116	IEIE Eila	01/19/22 16:20		root root		
mewcss.css	9.10	8 CSS Source File	01/22/22 22:47:09			wahah ing		65 031	IDG File	01/18/22 16:29	-TW-TT	root root		
newindex.ht	tml 7.54	6 Chrome HTML Do	01/18/22 15:36:39			Nonco.jpg	4	6 606 713	MP4 Video	01/18/22 16:29	-DW-Feefee	root root		
newiavascri	pt.is 1.82	4 JavaScript Source	01/18/22 01:46:58			Nideo4.m	4	14 985 417	MP4 Video	01/18/22 16:29:	-0	root root		
p1.jpeg	95,33	0 JPEG File	01/26/22 09:35:58			₩ video3.m	4	16.461.813	MP4 Video	01/18/22 15:45:	-1W-11	root root		
p2.jpeg	179,52	4 JPEG File	01/26/22 09:35:56		~	🔊 video2.mi	4	7 097 648	MP4 Video	01/18/22 15:45:	-DW-DD	root root		~
19 files. Total siz	ze: 69,658,533 bytes					18 files. Total	ize: 69,650,987 b	oytes						
Server/Local fil	le Direction	Remote file	Si	ize Priority	Status									

Figure 3.20: FileZilla

3.4.6 Twilio

Twilio is a customer engagement and interaction platform that is widely used in businesses that allowed to build of customized and unique messages for the customers such as promotions through message, voice, video, or email as shown in Figure 3.21. In this study, it is used to send an SMS alert to the smart laboratory authorities when there is a motion detected by the PIR motion sensor.

Console <u>My first Twilling</u>	accour	Trial: \$11.413 Upgrade		Q Jump to		Account ~	Billing 🗸	CE CARLOS ~
Develop Monitor		• Learn the basics of Twilio	How does Twilio Work?					×
 > [™] Studio [™] Functions		My first Twilio account 🖉 Hi the Check	ere! Want to get an app out our most popular use ca	running with no code?	See app	samples א		
 > C Messaging > S Voice 		Project Info						^
Explore Products	÷	ACCOUNT SID ACf1240b16836d02c70e01c6	57ddc22b12a	ă	TRIAL BALANCE			
Docs and Support	:	AUTH TOKEN Show		Ō				
	«	O Always store your token sec PHONE NUMBER	curely to protect your accoun	t. View best practices 🛪		Activate Go to Settin	Windows gs to activat	e Windows.

Figure 3.21: Twilio

3.5 System Architecture

In this study, the overall system architecture of the developed smart laboratory system is illustrated in Figure 3.22. A Wi-Fi-connected Raspberry Pi acts as a local server and the communication between clients and servers is using MQTT protocol to publish and subscribe the data. MQTT broker plays the role of receiving messages, filtering messages, and publishing the data to subscribed clients (Longo et al. 2020). In simple words, clients connect to the broker and subscribe or publish the data on specific topics. The Raspberry Pi will be used to process the data obtained from various sensors such as DHT11 temperature and humidity sensors and PIR motion sensors. All of the sensed data will be then collected, saved, and managed through Raspberry Pi. For data management, the sensors data will be locally stored in the MySQL database installed on Raspberry Pi. Node-RED provides the web interface and dashboard to control and monitor the sensors.

The ESP32 microcontroller with integrated Wi-Fi that is connected to the sensor performs the data collection from the connected devices. The temperature and humidity sensor updates the data readings to ESP32 and displays real-time readings on the Node-RED dashboard. The motion sensor collects the data and displays green when there is motion detected or stays red when there is no motion detected. The authorities receive notifications based on motion sensor detection. For example, an output signal is transmitted to ESP32 when the motion is detected, which triggers Twilio to send a message alert to the authorities. To control a relay module to switch the actuators with ESP32, a LOW signal is sent so that the current will flow to switch on the light bulb, while a HIGH signal is sent so that the current will stop flowing to switch off the light bulb. The ESP32-CAM displays the laboratory's real-time vision.



Figure 3.22: System Architecture

3.6 System Flowchart



Figure 3.23: System Flowchart

Figure 3.23 illustrates the conceptual flowchart of the smart laboratory system. First, the Raspberry Pi acts as the local server that is connected to Wi-Fi. ESP32 microcontroller that is attached with sensors is connected with Raspberry Pi. Each of the relays and the sensors are connected with an ESP32 microcontroller which is acting as MQTT clients communicating with Raspberry pi acting as MQTT broker.

MQTT allows sending a command to control output as well as publish and subscribe the data. For example, The Node-RED publishes messages such as activating

or deactivating the relays or actuators in the topic ESP32. The ESP32 is subscribed to the same topic. Then, it receives the message whether to activate or deactivate, and finally, the task of turning on or off the LED light bulbs will then be done. The data will be stored in the MySQL database. All of the monitoring systems data can be monitored and controlled via the smart laboratory system website with a cloud IP address that is established with DigitalOcean.

3.7 Product Testing

Before proceeding to product testing, the connection of the microcontroller with components and compiled programming codes need to be done to perform the required tasks in real-time testing. The purpose of product testing is to validate the effectiveness and functionalities of the smart laboratory system. Identifying and troubleshooting the problems should be carried on from time to time when there is failure detected in conducting the product testing. The product testing should be repeated several times until the finalized system can perform well and reaches the expected outcomes. At the same time, improvement and enhancement are also important to achieve the best result. The product testing is conducted in the CAD Laboratory of University Malaysia Pahang (UMP).

Figure 3.24 shows the first time getting the output readings that come from the DHT11 temperature and humidity sensor. It can detect the real-time temperature of laboratory surroundings after several times trying. The output reading is published and can be viewed on the Node-RED dashboard.



Figure 3.24: Testing of DHT11 Sensor with Output Readings

When testing for the PIR motion sensor, it is connected with a power bank so that it is more convenient to bring along from place to place for different views. It is then placed on the table, and the ESP32-CAM is facing the entrance of the laboratory as shown in Figure 3.25



Figure 3.25: Testing of ESP-CAM connected with Power Bank

Figure 3.26 shows that the camera is successfully displayed the real-time vision of the laboratory. From here, it is proved that the ESP32-CAM is functioning well.



Figure 3.26: Testing of ESP-CAM Display

A buzzer and a LED are added to the breadboard circuit to test the functionality of the PIR motion sensor, the circuit connection is shown in Figure 3.27. The buzzer should be triggered and the LED will be lighted up when there is movement detected by the motion sensor.



Figure 3.27: Circuit connection of PIR motion sensor

The functionality of the PIR motion sensor is tested two ways, one in a short distance and the other in a long distance. Figure 3.28 shows the motion detection of the motion sensor in a short distance. The testing is conducted by waving a hand in front of the motion sensor, the status of the motion sensor turned green, and the LED light up.



Figure 3.28: Testing PIR Motion Sensor in Short Distance

Next, motion detection for far distance is also tested as shown in Figure 3.29. This time, the buzzer is triggered and the LED lighted up when there is a human passed by the motion sensor, which means the motion is successfully detected even in a long distance. This can prove that the PIR motion sensor is functioning well. This step is repeated a few more times by adjusting the time delay as well as the sensitivity of the motion sensor. The adjustment might affect the capability of motion sensing and eventually leads to getting a different result.



Figure 3.29: Testing of PIR Motion Sensor in Long Distance

Figure 3.30 shows the circuit connection of relay modules and light bulbs. Both of the light bulbs can be lighted up when the switches are turned on from the Node-RED dashboard as shown in Figure 3.31.



Figure 3.30: Circuit Connection of Relay Modules and Light Bulbs



Figure 3.31: Testing for Controlling Light Bulbs

Throughout the process of product testing, all of the monitoring systems have been tested to prove that the selected components are functioning effectively connected to the ESP32 microcontroller. After the product testing, the real real-time will be displayed on the Node-RED dashboard. Users can always view the data anytime anywhere by accessing the developed web page of the smart laboratory system.

3.8 Cost Estimation

Cost analysis of the materials used for this project is discussed and considered carefully and unnecessary component is avoided. The price allocation and quantity needed are important in cost estimation. So, the components are well studied and selected to avoid future complications in the project.

No.	Items	Quantity	Price per unit	Price
1.	Raspberry Pi	1	RM 69.00	RM 69.00
2.	ESP32	4	RM 29.00	RM 116.00
3.	ESP32-CAM	1	RM 39.80	RM 39.80
4.	DHT11 Temperature and	2	RM 2.90	RM 5.80
	Humidity Sensor			
5.	PIR Motion Sensor	1	RM 15.90	RM 15.90
6.	Relay Module	2	RM 5.00	RM 10.00
7.	LED Light Bulb	2	RM 5.50	RM 11.00
8.	PCB Board	4	RM 9.00	RM 36.00
9.	Breadboard	2	RM 23.20	RM 46.40
10.	Electronic Project Box	1	RM 2.35	RM 2.35
	RM 352.25			

Table 2: Cost Analysis

RESULTS AND DISCUSSION

4.1 DigitalOcean Cloud Server Setup

Step 1 – Sign up Digital Ocean Account

DigitalOcean is used as a cloud server for a smart laboratory system in this study. The setup is easy with only a few simple steps. First, sign up for an account from DigitalOcean's official website, which is <u>https://www.digitalocean.com</u> as shown in Figure 4.1.



Figure 4.1: DigitalOcean Sign Up

Step 2 – Create Project

Once the account is successfully signed up, proceed with creating a new project at the side menu bar. Fill in the project name, description, and purpose of using DigitalOcean to create a new project as shown in Figure 4.2. In this study, the project name is set as Smart Laboratory System.

Create project - DigitalOcean X +
 C a cloud-digitalocean.com/projects/new7h=54380e Apps © WhatsApp V ECOMMUNITY - U V Knowledge and Lea. A How To Install the. S DD2 Technical Proj. S DD-FIXEE IS ESP32 MUTT Public. W HTML Tudoral Apps C a search by resource name or public IP (CtrH#) Create new project Name your project
Image: Image
Q. Search by resource name or public IP (CtrHB) PROJECTS W Train Create Create Create MANAGE Apps Mame your project Name your project
PROJECTS Apps
Image: Smart Labora Create new project MANAGE Apps Name your project
+ New Project MANAGE Apps New
Apps Kew Name your project
Apps New Name your project
Droplets Extension Smart Laboratory System
Kubernetes Add a description
Volumes Heipfuf for teams or differentiating between projects with similar names.
Databases Enter description
Spaces Toll us what it's for
Container Registry This will help us to provide a more relevant experience.
Images Class project / Educational purposes • ~
Networking
Monitoring Create Project
Settions

Figure 4.2: Create New Project at DigitalOcean

Step 3 – Create Droplet

Once the project has been successfully created, create a droplet, also known as a cloud server for the project. In this study, Ubuntu 20.04 (LTS) x64 is used as the server, a basic plan for shared CPU is selected as shown in Figure 4.3.

Initial Server Setup with	Ubuntu 🗧 🗙 , Create Droplets - Digita	Ocean × +				~ - 0 ×		
\leftrightarrow \rightarrow C \bigcirc clou	d.digitalocean.com/droplets/new?i=949	980e&size=s-4vcpu-8gb-amd&re	gion=sfo3		07	९ 🖻 🖈 🗈 💧 :		
🗰 Apps 🔇 WhatsApp	😲 E-COMMUNITY - U 🦞 Knowledge	and Lea 🥥 How To Install the	SDP2 Technical Proj	SDP-FTKEE R ESP32 M	QTT Publis 📢 HTML Tu	torial 🔹 🗎 Reading list		
ລ	 Q Search by resource name or put 	olic IP (Ctrl+B)		Crea	te∨ ⑦ ₽ 0	My Team redit: \$84.55 (expires in 15 days)		
PROJECTS ^	Create Droplets							
Smart Labora	Choose an image 👔							
+ New Project	Distributions Container o	istributions Marketplace C	Lustom images					
MANAGE ^								
Apps NEW	- O		£	\bigcirc	(1) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			
Droplets	Ubuntu	FreeBSD	Fedora	Debian	CentOS	Rocky Linux		
Kubernetes	20.04 (LTS) x64 V	Select version V	Select version V	Select version V	Select version	✓ Select version ✓		
Volumes								
Databases								
Spaces								
Container Registry	Choose a plan					Help me choose 🗹		
Images	SHARED CPU			DEDICATED CPU				
Networking Monitoring	Basic	General Purpose	CPU-Optim	ized Memor	y-Optimized	Storage-Optimized		

Figure 4.3: Image and Plan

Figure 4.4 shows the selected CPU option, which is Premium AMD with NVMe SSD with the size of 2 GB/2 AMD CPUs, 60 GB NVMe SSDs, and 3 TB transfer. The subscription plan costs \$18 per month (\$0.027 per hour).

-	-					
 Initial Server Setup with I 	Ubuntu 🛛 🗙 😡 Create Droplets - Digita	alOcean × +				∨ - ⊡ ×
\leftrightarrow \rightarrow G \oplus cloud	I.digitalocean.com/droplets/new?i=94	980e&size=s-2vcpu-2gb-am	d®ion=sfo3		0- Q	🖻 🖈 🖻 🌲 🗐 💧 E
Apps 🧕 WhatsApp	V E-COMMUNITY - U V Knowledg	e and Lea 🔵 How To Install	the 😑 SDP2 Technical Proj	SDP-FTKEE R ESP32 N	IQTT Publis 📢 HTML Tutoria	N 🗄 Reading list
ຸລ	Q Search by resource name or pu	blic IP (Ctri+B)		Crea	ate v ? Q. Credi	My Team t: \$84.55 (expires in 15 days)
PROJECTS ^						
💋 Smart Labora						
🌈 Smart Labora	Basic virtual machines with and dev/test environments.	a mix of memory and compu	te resources. Best for small p	projects that can handle varia	ble levels of CPU performan	ce, like blogs, web apps
+ New Project						
	CPU options: Regula	r Intel Premium Ir	ntel NEW O Premi			
MANAGE ^	with 35t	with NVMe 3		vme 550		
Apps NEW	\$ 6 /mo \$0.009/hour	\$12/mo \$0.018/hour	\$18/mo \$0.027/hour	\$24/mo \$0.036/hour	\$48/mo \$0.071/hour	\$ 96 /mo \$0.143/hour
Droplets	1CB / LAND CDU	2 CB (1 AND CDU	2 CB / 2 AMD CDLM		R CR / A AND CDU A	10 CD / R AND CDUA
Kubernetes	25 GB NVMe SSDs	50 GB NVMe SSDs	60 GB NVMe SSDs	80 GB NVMe SSDs	160 GB NVMe SSDs	320 GB NVMe SSDs
Volumes	1000 GB transfer	2 TB transfer	3 TB transfer	4 TB transfer	5 TB transfer	6 TB transfer
Databases						
Spaces	Basic Droplet plans have	e replaced Standard Droplet	plans. Users who have Stand	lard Droplets using deprecat	ed plans can still create Drop	elets with those plans using
Container Registry	the API.					
Images	Fach Decelet also includes 6		tish is shored between all D			is always from 1 and 1
Networking	or try our price calculator.	ee outbound data transfer v	mich is shared between all D	ropiets each billing cycle. Int	ound bandwidth to Droplets	is always free. Learn more
Monitoring						
monitoring						

Figure 4.4: CPU Options

Figure 4.5 shows that the country of Singapore is selected as the data center region. Among all the country options, Singapore is the most appropriate option as it is nearest to the country of this study, which is Malaysia.

Initial Server Setup with Ubur	ntu 🗄 🗙 🔵 Create Droplets - Digital	Ocean × +				~ - @ ×		
\leftrightarrow \rightarrow C $($ cloud.dig	gitalocean.com/droplets/new?i=949	180e&size=s-2vcpu-2gb-am	d®ion=sgp1		0 7 Q	🖻 🖈 🖻 🛊 🗐 💧 :		
👯 Apps 🧕 WhatsApp 🎈	E-COMMUNITY - U 💙 Knowledge	and Lea 🥥 How To Install t	he 言 SDP2 Technical Proj	SDP-FTKEE R ESP32 M	IQTT Publis 📢 HTML Tutorial	» 🔝 Reading list		
୍	Q Search by resource name or pub	blic IP (Ctrl+B)		Crea	tev 🧿 🗘 Credit	My Team 💦 🕯		
PROJECTS ^	Add block storage 👔							
 Smart Labora + New Project	Add Volume							
MANAGE ^	Choose a datace	nter region						
Apps NEW								
Droplets				(0	×			
Kubernetes	New York	San Francisco	Amsterdam	Singapore	London	Frankfurt		
Volumes	1 2 3	1 2 3	2 3	1	1	1		
Databases								
Spaces	+	9						
Container Registry	Toronto	Bangalore						
lmages	1	1						
Networking								
Monitorina								

Figure 4.5: Datacenter Region

In Figure 4.6, a root password is created to access the droplet for authentication and security purposes. The password should be created according to the requirements stated below.

😡 Initial Server Setup with Ubu	tu x 💭 Create Droplets - DigitalOcean x + 🗸 - 🗗 🗧	×
\leftrightarrow \rightarrow C $$ cloud.dig	talocean.com/droplets/new?i=94980e&size=s-2vcpu-2gb-amd®ion=sgp1 🛛 🗣 🔍 🖻 🏚 🗐 👸	:
🛄 Apps 🧕 WhatsApp 😲	E-COMMUNITY - U V Knowledge and Lea 🔉 How To Install the 🗧 SDP2 Technical Proj 🖬 SDP-FTKEE 🔃 ESP32 MQTT Publis 📢 HTML Tutorial 🔹 🗎 Reading in	ist
୍ଚ 🌔	o, Search by resource name or public IP (CtrHB) Oreate V 🕜 🖨 Create States (corputs in 15 days)	*
PROJECTS ^	default-sgp1 DEFAULT	
💋 Smart Labora	All resources created in this datacenter will be members of the same VPC network. They can	
🌈 Smart Labora	communicate securely over their Private IP addresses. What does this mean?	
+ New Project		
MANAGE ^	Authentication 3	
Apps NEW Droplets	SSH keys A more secure authentication method Password Create a root password to access Droplet (less secure)	i.
Kubernetes		
Volumes		
Databases		
Spaces	The ten besterior.	
Container Registry	PASSWORD REQUIREMENTS	
Images	Must be at least 8 characters long Must contain 1 uppercase letter (cannot be first or last character)	
Networking	Must contain 1 number	
Monitoring	Cannot end in a number or special character	
	Please store your password securely. You will not be sent an email containing the Droplet's details or password.	-

Figure 4.6: Authentication

After completing the droplet, an IP address for the cloud server will be created. For this project, the IP address is 178.128.50.80 as shown in Figure 4.7

Initial Server Setup with	h Ubuntu 🗧 🗙	💭 Smart La	aboratory System project 🗙 🕂					~	- 0 ×
\leftrightarrow \rightarrow C \cong clou	ud.digitalocea	n.com/projec	cts/dbadb5db-4f9e-4ded-aa38-6119b4	314836/resources?	=94980e			• • • •	🗈 A 🙆 E
🗰 Apps 😒 WhatsApp	U E-COMM	IUNITY - U	😲 Knowledge and Lea 🕥 How To Ins	tall the 📑 SDP2 '	Technical Proj 🖬	SDP-FTKEE 🛛 ESP32 MC	TT Publis 📢 HTML To	utorial	» 🗄 Reading list
ລ	Q Sear	ch by resourc	ce name or public IP (Ctrl+B)			Create	× 0 Û	Credit: \$84.55 (expires	My Team in 15 days)
PROJECTS ^									
🌽 Smart Labor			mart Laboratory Syst	OM DEFAULT					
+ New Project		c	lass project / Educational purposes / U	Ipdate your project	information under	Settings		→ Move F	Resources
MANAGE ^	Re	sources /	Activity Settings						
Apps NEW	DB	OPLETS (1)							
Droplets	DR	OFLETS (I)							
Kubernetes			rtiab						· ···
Volumes		Silia	ruad						
Databases									
Spaces		Image	Obuntu 20.04 (LTS) x64	Region	SGP1				
Container Registry		Size	2 vCPUs 2GR / 60GR Disk	IPv4	178.128.50.80				
Images			(\$18/mo)	IPv6	Enable				
Networking			Resize	Private IP	10.104.0.2				
Monitoring				W.C	deladiesgpr				

Figure 4.7: Complete Creating Droplet

4.2 Apache and FileZilla Installation

Step 1 – Login as Root

Since the cloud server has been successfully created, an Apache web server is needed to be installed on the Ubuntu 20.04 server. Before the installation starts, log in to the server as root using the IP address and password set previously in DigitalOcean. By using the command of ssh <u>root@178.128.50.80</u>, enter the password as shown in Figure 4.8.

```
C:\Users\user>ssh root@178.128.50.80
root@178.128.50.80's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-100-generic x86 64)
  Documentation: https://help.ubuntu.com
Management: https://landscape.canonical.com
Support: https://ubuntu.com/advantage
 * Support:
 System information as of Tue Feb 22 04:27:50 UTC 2022
 System load: 0.0
                                       Users logged in:
                                                                   Ø
                 10.1% of 57.98GB IPv4 address for eth0: 178.128.50.80
  Usage of /:
                                        IPv4 address for eth0: 10.15.0.5
 Memory usage: 22%
  Swap usage:
                 0%
                                        IPv4 address for eth1: 10.104.0.2
  Processes:
                  166
```

Figure 4.8: Log In as Root

Step 2 – Install Apache

Apache webserver function is to display the developed web page for the smart laboratory system. To install Apache, use the commands to run as below:

\$sudo apt-get update

\$sudo apt-get install apache2

\$sudo ufw app list

\$sudo ufw allow 'Apache'

Once it is successfully installed and after running to check the status, it should be showing "Active" as shown in the textbox below.

Status: active		
То	Action	From
Apache	ALLOW	Anywhere
Apache (v6)	ALLOW	Anywhere (v6)

Now, try to access the server's IP address of 178.128.50.80 and it will display the default Ubuntu 20.04 Apache web page as shown in Figure 4.9.

Image: style It works! This is the default welcome page used to test the corect operation of the Apache2 server after installation on Uburn systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace to the (located at /var/ww/html / Index. Intel) before continuing to operate your HTTP server. If you are a normal user of this web site and dor't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem presists, please contact the site's administrator. Uburuty: A pache2 default configuration is different from the upstream default configuration, and split into several file optimized for interaction with Ubuntu tools. The configuration for the wild occumentation. Documentation for the web server itself can be found by accessing the manual if the apache2 / doc package was installed on this server. If configuration layout for an Apache2 web server installation on Uburut systems is as follows: /etc/apache2/ - apache2.conf ports.conf conf		V
It works1 This is the default welcome page used to test the correct operation of the Apache2 server after installation on Uburni systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you ead this page, it means that the Apache HTTP server installed at this site is working properly. You should replace to the (located at / var/www/html.) index. Inthis before continuing to operate your HTTP server. If U coated at / var/www/html.) index. Inthis before continuing to operate your HTTP server. If unavailable due to maintenance. If the problem persists, please contact the site's administrator. Configuration of Var/Www/html. Yourkow. Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several file usershare/docapache2/README.Debiang. Refer to this for the full documentation. Documentation for the web usershare/docapache2/README.Debiang. Refer to this for the full documentation. Usershare/docApache2/README.Debiang. Refer to this for the full documentation. - apache2.conf - apache2.conf - apache2.conf - enabled s.conf - seconf - seconf - seconf - apache2.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime. Configuration files in the mods - enabled/, conf - enabled/ at sites - enabled/ directories contain particular configuration sinppets which manage modules, global configuration finguration, apache2 nedis		ubuntu
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubunt systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you ead this page, it means that the Apache HTTP server installed at this site is working properly. You should replace to the (located at /var/www/html/indv.html) before continuing to operate your HTTP server. Tyou are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator. Configuration Overview Dutut's Apache2 default configuration is different from the upstream default configuration, and split into several file profinized for interaction with Ubuntu tools. The configuration system is fully documented in usrfshare/docApache2/README.Debiang. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server. The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows: /etc/apache2/ apache2.conf apache2.conf sites -enabled sites -enabled sites -enabled sites -enabled so file always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime. Configuration files in the mods -enabled/, configuration files from their respective *-available/ counterparts. These should be manage to building and sites -enabled/ directories contain particular configuration sinpers which manage modules, global configuration files, not available counterparts. These should be managed by using our helpers azemod , azdissind , azensite , azdissite , and azenconf , azdisconf . See their respective ma		It works!
space 2 configuration overview Durut's Apache2 default configuration system is fully documented in usr/share/doc/apache2/README_Debina.gz. Refer to this for the full documentation. Documentation for the web rever itself can be found by accessing the manual if the apache2 - doc package was installed on this server. /etc./apache2/ / apache2.conf apache2.conf apache2.conf apache2.conf apache2.conf sites-enabled *.conf apache2.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled *.conf sites-enabled words-enabled *.conf sites-enabled *.conf sites-enabled/ *.conf words-enabled/ *.conf	This syste read ile (f you	is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubunt ms. It is based on the equivalent page on Debian, from which the Ubunt Apache packaging is derived. If you c this page, it means that the Apache HTTP server installed at this site is working properly. You should replace tl located at /var/www/html/index.html) before continuing to operate your HTTP server. are a normal user of this web site and don't know what this page is about, this probably means that the site is
Configuration Overview Jbuntu's Apache2 default configuration is different from the upstream default configuration is different from the upstream default configuration for the web prever itself can be found by accessing the manual if the apache2-doc package was installed on this server. The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows: /etc/apache2/	curre	ently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Jbuntu's Apache2 default configuration is different from the upstream default configuration, and split into several file pilmized for interaction with Ubuntu tools. The configuration system is fully documented in usr/share/doc/apache2/README_Deblan.gz. Refer to this for the full documentation. Documentation for the web erver itself can be found by accessing the manual if the apache2-doc package was installed on this server. The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows: /etc/apache2/ i apache2.conf ports.conf mods-enabled *.conf store apache2.conf store apache2.conf conf-enabled *.conf *.conf store store *.conf store *.conf <td></td> <td>Configuration Overview</td>		Configuration Overview
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows: /etc/apache2/ apache2.conf apache2.conf mods-enabled *.load *.conf sites-enabled sites-enabled *.conf sites-ena	Jbur optin usr/	tut's Apache2 default configuration is different from the upstream default configuration, and split into several file nized for interaction with Ubuntu tools. The configuration system is fully documented in share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web er itself can be found by accessing the manual if the apache2-doc package was installed on this server.
<pre>/etc/apache2/ apache2.conf mods-enabled</pre>	he	configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
 apache 2. conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server. ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime. Configuration files in the mods - enabled/, conf - enabled/ and sites - enabled/ directories contain particular configuration simplets which manage modules, global configuration fragments, or virtual host configurations, respectively. They are activated by symilnking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers aZemmod, aZdismod, aZensite, aZdissite, and aZenconf, aZdisconf .See their respective man pages for detailed information. The biany is called apache2. Due to the use of environment variables, in the default configuration, apache2 meeds to be started/stopped with /etc/init.d/apache2 or apache2ctl.Calling /usr/bin/apache2 directly will not work with the default configuration. Bocument Roots By default, Ubuntu does not allow access through the web browser to any file apart of those located in /var /www, tublic. Itml directories (when enabled) and /usr/share (for web applications). If your site is using a web docume tool tocated deswhere (such as in /srv) you may need to whitelist your document root directory in ^{et}ct/apache2/apache2.conf. the default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This i different to previous releases which provides better security out of the box. Reporting Problems Pease use the ubuntu- bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug eports before reporting a new bug. 	 	Labella Conf ports.conf mods-enabled [*.load *.conf - sites-enabled *.conf
 ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime. Configuration files in the mods -enabled/, conf -enabled/ and sites -enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively. They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers aZemmod, aZdismod, aZensite, aZdissite, and aZenconf, aZdisconf. See their respective mapages for detailed information. The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 meeds to be started/stopped with /etc/init.d/apache2 or apache2ctl.Calling /usr/bin/apache2 directly will not work with the default configuration. Document Roots y default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www. Nublic.html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume to clocated desewhere (such as in /srv) you may need to whitelist your document root is /var/www/.html. You can make your own virtual hosts under /var/www. This i lifterent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu -bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug ports before reporting a new bug. 		apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
 Configuration files in the mods - enabled/, conf - enabled/ and sites - enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively. They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite, a2dissite, and a2encoff, a2discoff. See their respective man pages for detailed information. The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration. Document Roots 2y default, Ubuntu does not allow access through the web browser to any file apart of those located in /var /www, public, html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume oot located elsewhere (such as in /srv) you may need to whitelist your document or directory in fet of pache2. a conf. He default Ubuntu document root is /var/www/.html. You can make your own virtual hosts under /var/www. This i lifferent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu- bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug eports before reporting a new bug. 		ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers aZenmod, aZdismod, aZensite, aZdissite, and aZenconf, aZdisconf. See their respective man pages for detailed information. The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 meds to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration. Document Roots y default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www, sublic_html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume oot located elsewhere (such as in /srv) you may need to whitelist your document root directory in fetc/apache2/apache2.conf. the default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This is lifferent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu- bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug ports before reporting a new bug.		Configuration files in the mods -enabled/, conf -enabled/ and sites -enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be stated/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration. Document Roots 3 default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www, bublic_html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume oot located elsewhere (such as in /srv) you may need to whitelist your document root directory in fet c/apache2/apache2.conf. the default Ubuntu doces not on is /srv/you may need to whitelist your document root directory in fet c/apache2/apache2.conf. the default Ubuntu document root is /var/www.html. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bus ports before reporting a new bug.	•	They are activated by symlinking available configuration files from their respective +available/ counterparts. These should be managed by using our helpers a2enmod , a2dismod , a2ensite , a2dissite , and a2encorf , a2disconf . See their respective man pages for detailed information.
Document Roots by default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www, ublic, html directories (when enabled) and /usr/share (for web applications). If your is is using a web docume oot located elsewhere (such as in /srv) you may need to whitelist your document root directory in fet c/apache2/apache2.conf. the default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This i lifferent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug eports before reporting a new bug.		The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration.
by default, Ubuntu does not allow access through the web browser to any file apart of those located in /var/www, buble_html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume to located deswhere (such as in /srv) your may need to whitelist your document root directory in fetc/apache2/apache2.conf. the default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This i lifferent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug ports before reporting a new bug.		Document Roots
The default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This i illerent to previous releases which provides better security out of the box. Reporting Problems Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug eports before reporting a new bug.	By d bubl oot	efault, Ubuntu does not allow access through the web browser to <i>any</i> file apart of those located in /var/www, ic_html directories (when enabled) and /usr/share (for web applications). If your site is using a web docume located elsewhere (such as in /srv) you may need to whitelist your document root directory in c/apache2/apache2.conf.
Reporting Problems Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing buy eports before reporting a new bug.	The liffer	default Ubuntu document root is /var/www/html. You can make your own virtual hosts under /var/www. This i rent to previous releases which provides better security out of the box.
Please use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug eports before reporting a new bug.		Reporting Problems
	epo	se use the ubuntu-bug tool to report bugs in the Apache2 package with Ubuntu. However, check existing bug

Figure 4.9: Default Ubuntu Apache Web Page

Step 3 – Install FileZilla

In the command prompt, install FileZilla by running the command stated in the textbox below:

\$sudo apt-get update

\$sudo apt-get install filezilla

4.3 FileZilla File Transfer

After installing FileZilla in the Ubuntu server, open the FileZilla application on the desktop and connect to the server to do the work of files transfer. Figure 4.10 shows the FileZilla overview. The steps of file transferring are stated as below:

- 1. **Quick connect:** It is used to connect to the hosting server quickly by entering host, username, password, and port. In this study, the host is sftp://178.128.50.80, the username is "root", the password for the server is entered, connect using port 22.
- 2. **Message log:** The message log shows the connection status, command, and response when there are any actions taken by the user to show whether the server is connected. The file transfer status will also be shown here.
- 3. Local Site: On the left is called Local Site, all the folders of a selected path on the local system will be shown. The selected files can be dragged from the left to right side to transfer from the local server to the cloud server.
- 4. **Remote Side**: On the right is called Remote Site. This section will be showing all the files that have been transferred from the local server. The dropped files can be deleted. In this study, all the files such as coding, images, and videos that were used in developing the web page have been transferred to the remote site with the path of /var/www/html.

E sftp://root@178.128.50.80	FileZilla						-	٥	×
File Edit View Transfer	Server Bookmarks Help								
₩ • • • • • •	🖸 🏦 🛛 🐛 🐌 🎞	Q 🗢 🧥							
Host: ftp://178.128.50.80 U	Jsername: root	Password: Port: 22	Quickconne	ct 🕶 1					
Status: Retrieving directory Status: Listing directory / Status: Directory listing or Status: Listing directory Status: Listing directory Status: Listing directory Status: Directory listing or Status: Directory listing or Status: Directory listing or Status: Directory listing or	ry listing of "/var" var f "/var" successful ry listing of "/var/www" var/www t "/var/www" successful ry listing of "/var/www/html"	2							^
Status: Directory listing o	f "/var/www/ntml" successful								
									~
Local site: C:\Users\user\Des	ktop\UNI\Year 4 Sem 1\SDP2\W	ebpage\new try\	~	Remote site: /var/www/	html				~
	Technical Report	3	~	? opt ? run ? snap ? spool ? tmp	4				^
Filename	Filesize Filetype	Last modified	^	ė- 🔒 www				- I	- 1
p2.jpeg	179,524 JPEG File	01/26/22 09:35:56		i 🔥 html					¥
🖬 p3.jpeg	90,161 JPEG File	01/26/22 09:36:06		Filename	Filesize Filetype	Last modified Permissions	Owner/Group		^
🖻 p4.jpeg	159,769 JPEG File	01/26/22 09:36:09		. .				- I	
🔊 video1.mp4	23,643,485 MP4 Video	01/16/22 16:03:15		🖻 yiming.jfif	6,116 JFIF File	01/18/22 16:29:rw-rr	root root	- I	
💐 video2.mp4	7,097,648 MP4 Video	01/16/22 15:59:45		🖬 waheb.jpg	65,031 JPG File	01/18/22 16:29:rw-rr	root root	- I	
Nideo3.mp4	16,461,813 MP4 Video	01/16/22 16:04:44		🔊 video5.mp4	6,606,713 MP4 Video	01/18/22 16:29:rw-rr	root root	- I	
😰 video4.mp4	14,985,417 MP4 Video	01/16/22 16:00:10		🔊 video4.mp4	14,985,417 MP4 Video	01/18/22 16:29:rw-rr	root root	- I	
Video5.mp4	6,606,713 MP4 Video	01/16/22 16:05:50		🔊 video3.mp4	16,461,813 MP4 Video	01/18/22 15:45:rw-rr	root root	- I	
waneb.jpg	05,031 JPG File	01/16/22 14:10:51		🔊 video2.mp4	7,097,648 MP4 Video	01/18/22 15:45:rw-rr	root root	- I	
🔤 yiming.jtir	0,110 JFIFFIIE	01/16/22 13:59:12	×	🔊 video1.mp4	23,643,485 MP4 Video	01/18/22 15:45:rw-rr	root root	- I	~
19 files. Total size: 69,658,533 b	ytes			18 files. Total size: 69,650,9	87 bytes	01/06/00.00.00			
Server/Local file	Direction Remote file	Size Priority Sta	atus						
Queued files Failed transf	ers Successful transfers								

Figure 4.10: FileZilla Overview

4.4 Development of Web Page

A few programming languages were used in developing a web page for the smart laboratory system, which included PHP Hypertext Preprocessor (PHP), Cascading Style Sheets (CSS), and JavaScript. PHP is widely used as a server-side scripting language. CSS is a language that describes how to display and style HTML documents so that it looks more presentable. JavaScript is often used to enhance and add functions to a web page such as buttons to create animations or content.

4.4.1 PHP

Section <head>: In this section, all the connected files in developing the web page are inserted inside the <head> tag such as CSS files, language, web page title as shown in Figure 4.11. The file name or the path for <link rel= "stylesheet" href= ""> must be entered correctly, or else it will not link with the specific files hence it will not display any styles for the web page.



Figure 4.11: PHP <head>

Section <body>: Inside the <body> tag, it starts with the header for the web page. There are several <div> tags that were used to separate the section of the HTML document. A navigation bar is created for users to select the web page topics easily, which are 'Home', 'Services', 'Gallery', and 'About Us' as shown in Figure 4.12. The will be directing the user to the destination when they click the topic. Besides, a logo was added on the left side of the navigation bar by inserting the logo image using the tag. An input type is used in this section which is <input type= "search"> for the users to search anything on the developed web page.



Figure 4.12: PHP <body>

Section Login Form: A login form is required in the future study so that only members/students/staffs are allowed to log in to the smart laboratory web page just to view the real-time data for more privacy purposes. The outsider can only view the basic data or information such as temperature readings in the smart laboratory. In this division, there is a total of four input types for the login form, which are email, password, submit, and checkbox as shown in Figure 4.13.

<input type= "email">: used for e-mail address input filed.

<input type= "password">: to insert password with shown in asterisks or circles.

<input type= "submit">: to submit the input data.

<input type= "checkbox">: enable the user to tick the options for one or more.

48	
49	login form container
50	
51	<div class="login-form-container"></div>
52	
53	<i class="fas fa-times" id="form-close"></i>
54	
55	<form action=""></form>
56	<h3>login</h3>
57	<input class="box" placeholder="enter your email" type="email"/>
58	<input class="box" placeholder="enter your password" type="password"/>
59	<input class="btn" type="submit" value="login now"/>
60	<input id="remember" type="checkbox"/>
61	<label for="remember">remember me</label>
62	forget password? click here
63	don't have and account? register now
64	
65	
66	

Figure 4.13: PHP Login Form

Section Home: In this section, a large title to catch the user's attention is created by using the <h1> header tag named "UMP Smart Laboratory". Another line with a <h3> tag, which is a smaller header is used to display the simple description. After this, a button is used to allow users to view the real-time data display on the Node-RED dashboard. Hence, the IP address for the Node-RED dashboard, which is http://192.168.2.127:1880/ui, is inserted in the <a href> tag to direct the user to view the data as shown in Figure 4.14. A few videos are used for users to have a clearer view of the laboratory concept, but it is just a style in this web page study.

```
68 <!-- home section starts -->
69
6
68 <!-- home section class="home" id="home">
71
73 {
74 
74 
75 
76 
77 
77 
78 
79 
79 
70 
70 
70 
70 
70 
71 
71 
72 
73 
74 
74 
75 
76 
76 
77 
78 
79 
79 
70 
70 
70 
70 
70 
71 
70 
71 
71 
72 
73 
74 
74 
75 
76 
76 
77 
76 
78 
79 
79 
70 
70 
70 
70 
70 
70 
70 
70 
71 
71 
72 
73 
74 
74 
75 
75 
76 
76 
77 
76 
77 
78 
79 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70 
70
```

Figure 4.14: PHP Home Section

Section Services: Figure 4.15 shows the <section class= "services">, the h1 heading is used to show the topic of the services. The <div class= "box-container"> is used to locate the boxes of services content. In this web page development, there is a total of four boxes showing the services provided by the smart laboratory system. Each of the content of the services has an appropriate icon for easy understanding. The name for each of the services is styled by using the h3 heading, with normal font size for description at the next paragraph of the title.



Figure 4.15: PHP Services Section

Section Gallery: Figure 4.16 and Figure 4.17Figure 4.15 shows the gallery section, which is to display some photos of the hardware system that was used in this project. Same with the services section, the h1 heading is used to show the gallery topic and the <div class= "box-container"> is used to locate the boxes of the gallery content. Four images have been put in the <div class= "box"> and displayed on the web page by using the <imp src>. The gallery content is styled with h3 headings for the gallery title and normal font size for the gallery description.



Figure 4.16: PHP Gallery Section 1



Figure 4.17: PHP Gallery Section 2

Section About Us: Figure 4.18 and Figure 4.19 shows the scripting for this section. The h1 heading is used to show the about us topic. In the <div class="swiper-container about-slider"> it is style by having the ability to swipe and view the about us content. In the <div class="swiper-wrapper">, there are five <div class="swiper-slide"> and <div class="swiper-slide"> to include all the project coordinator, supervisor and members simple information.



Figure 4.18: PHP About Us Section 1



Figure 4.19: PHP About Us Section 2

Section Footer: The footer of the web page is the social media account such as Facebook, Instagram, and WhatsApp as shown in Figure 4.20. It is convenient for people to have updates on our smart laboratory. Besides, it is easier for them to contact us for any inquiry. A JavaScript scripting is placed in using the <script> tag and link to the file for some functions in the developed web page.



Figure 4.20: PHP Footer

4.4.2 CSS

Section for overall: As shown in Figure 4.21, the dark cyan color is used as the root color of the designed web page. While 'Verdana', sans-serif is used as the font family. The html and section are simply styled with the 55% for font size and align all the items at the center of the web page.

For the topic headings as shown in Figure 4.22, they are all in dark cyan color and transformed into uppercase, bolder, and extra-large size with a letter spacing of 3px. A light grey text shadow with 2px is added to style the headings. For the button, the background is dark cyan color with bold white text, a solid white border. The cursor used is a pointer so that the user understands that the button can be clicked. The button effects are added to style with the use of the :hover selector and :active selector. For :hover selector, when the user mouse over the button, the button background turned from dark cyan to white, while the text will be turned from white to dark cyan. For :active selector, when the user clicked on the button, a grey box-shadow appeared with an effect of "pressed", which is transformed 4px for them to know that the button is clicked.



Figure 4.21: CSS Overall Section 1



Figure 4.22: CSS Overall Section 2

Section Header:

• Navigation Bar: Figure 4.23 shows the scripting for the grey navigation bar of the web page, also known as "navbar" in the scripting. A logo is placed on the left side of the navbar. There are four topics displayed in the navbar, space between is used to separate all of the topics so that it looks more presentable. Since the navbar background is grey, the color chosen for the text is white. When the user mouse over it, the text color changed from white to dark cyan. The color changes are the same applied to the icon of the search bar and login form at the right side of the navbar.



Figure 4.23: CSS Header Navigation Bar

• Search Bar: Figure 4.24 shows the search bar on the navbar. It allows the user to search anything on the web page. The flex is used to display an element as a block-level flex container, and all items are aligned at the center. The display of clip-path polygon property is used to style the search bar container. The search bar width is set to 100% with 1rem padding and 0.5rem border-radius to have a rounded corner of the search bar.



Figure 4.24: CSS Header Search Bar

• Login Form: The login form scripting is shown in Figure 4.25, Figure 4.26, Figure 4.27, and Figure 4.28. It is styled with a white background with a width of 50rem, and a 0.5rem border-radius to have a slightly rounded corner. The h3 heading is applied to the title of the login form with transformed uppercase text.

130	header .search-bar-container label{
131	color: #fff;
132	cursor: pointer;
133	font-size: 3rem;
134	margin-left: 1.5rem;
135	
136	
137	header .search-bar-container label:hover{
138	color:var(darkcyan);
139	
140	
141	.login-form-container{
142	position: fixed;
143	top:-120%; left: 0;
144	z-index: 10000;
145	min-height: 100vh;
146	width:180%;
147	background:□rgba(0,0,0,.7);
148	display: flex;
149	align-items: center;
150	justify-content: center;
151	
152	
153	.login-form-container.active{
154	top:0;
155	
156	

Figure 4.25: CSS Header Login Form Section 1



Figure 4.26: CSS Header Login Form Section 2



Figure 4.27: CSS Header Login Form Section 3



Figure 4.28: CSS Header Login Form Section 4

Section Home: The Home section of scripting is shown in Figure 4.29, Figure 4.30, and Figure 4.31. All of the contents of the home are placed at the center with a height of 100vh, which will be displayed on the whole screen. The tile of "UMP Smart Laboratory" is styled by using the h1 heading, uppercase text-transform with 30px font size, and added the text-shadow. The small description at the next line is designed with a smaller font size, lowercase text-transform, and added text-shadow Both the title and description text is in white.

Besides, a few videos are displayed on the home page to make the web page more vivid and attractive. The videos are placed with 100% width and height, the cover is selected as the object-fit. Users can control and watch different videos by clicking the round video buttons.







Figure 4.30: CSS Home Section 2



Figure 4.31: CSS Home Section 3

Section Services: The services scripting is shown in Figure 4.32 and Figure 4.33. A box container with a gap of 1.8rem is used to fit the boxes to show the services provided by the smart laboratory. There are four dark cyan boxes of services content, with a border radius of 1.5rem and padding of 3rem at the top and bottom, 2rem at the left and right side. The appropriate icons are used at each service provided with a wheat color, which is the same color as the h3 headings services title. While the description of the services is styled by using white text color aligned at the center, with smaller font size. The :hover selector is also used in the services section. When the user mouse over each of the boxes, it will have a bigger scale with a grey color box-shadow so that the user can read the content easily.



Figure 4.32: CSS Services Section 1



Figure 4.33: CSS Services Section 2

Section Services: The gallery scripting are shown in Figure 4.34 and Figure 4.35. The gallery is to show some of the prototype images of the project. A box container with a gap of 1.5rem is used to fit the boxes to show the photos. There are four 30rem height boxes of the gallery with 1rem of the solid white border and 0.5rem border-radius. The images are displayed with 100% of height and width, the cover is selected as the object-fit. Users can view the image description content only when they mouse
over the boxes. The content with wheat color title and smaller white color font size will flow from the top of the box to the bottom. The content is aligned at the center of gallery boxes.



Figure 4.34: CSS Gallery Section 1



Figure 4.35: CSS Gallery Section 2

Section About Us: A slider is used to style the introduction of the project coordinator, supervisor, and members. The scripting is as shown in Figure 4.36. There are five light grey boxes in this section, with a height of 40vh and 5px of border-radius. They are also styled with box-shadow, 1rem of padding with items aligned at the center. The profile image is set to 15rem for height and width. The border-radius is set to 50% so that it has a circular display for the image. The margin at the top and bottom are both set in 2rem and the cover is chosen as the object-fit. The project position is styled with a dark slate grey color and transformed the text to bold uppercase with the size of 2rem. The name and faculty of each member are simply styled with black color and a smaller

font size, which is 1.5rem. The user can scroll to the left or right to view the profile content.



Figure 4.36: CSS About Us Section

Section Footer: This section's scripting is as shown in Figure 4.37 and Figure 4.38. Footer is used to directing the users to follow us on social media such as Facebook, Instagram, and WhatsApp. The footer background color is slightly lighter grey than the navbar background color. The content in this container is justified to center. The heading of "Follow Us" is styled by using h3 heading with padding of 0.5rem.

Each social media icon is placed side by side with a gap of 20px and the appropriate color is applied to each of them. When the user mouse over the social media, the icon and text will have a bigger scale. At the same time, the icon and text of Facebook will turn blue color, Instagram will turn tomato color, and WhatsApp will turn green color.







Figure 4.38: CSS Footer Section 2

Media queries: Figure 4.39, Figure 4.40, and Figure 4.41 show the scripting of media queries that are used to create a different layout for different screen sizes for different devices. Since the smart laboratory system is allowed the users to control and monitor the laboratory appliances wirelessly and at a distance, it is better to have a proper display for smartphone devices. There are several different sizes are developed in this web page such as 1200px, 991px, 768px, and 450px. There will be slight changes according to the sizes so that every different device can have a proper display of the user interface web page.

479	
480	
481	@media (max-width:1200px){
482	
483	html{
484	font-size: 55%;
485	
486	
487	
488	
489	@media (max-width:991px){
490	
491	header{
492	padding:2rem;
493	
494	
495	section{
496	padding:2rem;
497	
498	
499	
500	
501	@media (max-width:768px){
502	
503	#menu-bar{
504	display: initial;
505	









Figure 4.41: CSS Media Queries Section 3

4.4.3 JavaScript

As shown in Figure 4.42, the document.querySelector() method returns the first element that matches a CSS selector. It can be the id selector or the class selector. When the window on scrolls, the search bar, and the login form will disappear if they are previously opened. The addEventListener() method attaches an event handler to an element. It is often applied on adding a click event to an <button> element such as search button, login button, close button, and video button.



Figure 4.42: JavaScript Section 1

The scripting for the slider effect of the about us section is shown in Figure 4.43. The space between each box is set to 50, autoplay with a delay of 2800. There are 3 breakpoints which are 640px can only view one slide, 768px can view 2 slides, and 1024px can view 3 slides.



Figure 4.43: JavaScript Section 2

4.5 Project Outcome

A user-friendly and simple UI interface web page has been developed using PHP, CSS, and JavaScript. The Node-RED dashboard link is inserted in the PHP scripting by using the <a> element with the href attribute, which indicates the link's destination. As shown in Figure 4.44, when the user clicks on the "Click to View" button, they will be directed to the destination, which is the Node-RED dashboard to view the laboratory's data or information. Users can also click the video buttons to view different videos.



Figure 4.44: Smart Laboratory System Home Web Page

Figure 4.45 shows when the search button with a magnifying glass icon is pressed. Users can type anything they wanted to search from the web page. The cross icon at the navbar can be clicked if the user wanted to close the search bar.



Figure 4.45: Search Bar Outcome

Figure 4.46 shows the login form. Users are required to enter their email address and password to log in. Also, the user can tick the checkbox of "remember me" so that it is convenient for login next time. Users can click to reset the password if they forgot their login password, and they can register an account if they do not own one.



Figure 4.46: Login Form Outcome

Figure 4.47 shows the services provided by the smart laboratory system. A simple description for each service is added.

SERVICES



Figure 4.47: Web Page Part Services

Figure 4.48 shows the gallery image's outcome. The content description has appeared once the user mouse over the specific image.

GALLERY



Figure 4.48: Web Page Part Gallery

Figure 4.49 shows the about us section of the web page. It is a simple and easy understanding profile introduction with a clear background and profile image to show the professional of member's name and position. Users can slide to the left or right to view more content.



Figure 4.49: Web Page Part About Us

Figure 4.50 shows the footer of the web page which directs the users to follow us on social media. The color change can be seen when the user mouse over it.



A side navigation bar at the left side displays all the laboratory data which included the temperature and humidity readings for both area 1 and area 2, camera display, motion detection status, and switches for the lamp as shown in Figure 4.51. Users can select to view any one of the laboratory's latest details by just clicking the sidebar menu.



Figure 4.51: Side navigation bar

While testing the smart laboratory system in this project, 2 units of DHT11 temperature and humidity sensors were placed at 2 different areas of the laboratory. For area 1, which is the front side of the laboratory, 2 air conditioners have been turned on. On the other hand, at the backside of the laboratory which is area 2, the air conditioners have been turned off. From this situation, area 2 will have a higher temperature and lower humidity compared to area 1. The results can be seen in both Figure 4.52 and Figure 4.53Figure 4.53.



Figure 4.52: Temperature and Humidity in Area 1 of Laboratory



Figure 4.53: Temperature and Humidity in Area 2 of Laboratory

The ESP32-CAM function displays the real-time laboratory situation as shown in Figure 4.54. The users can view the real-time vision of the laboratory anytime and anywhere with a wireless connection by just accessing the smart laboratory website.



Figure 4.54: ESP32-Camera Display

The PIR motion sensor is placed at the laboratory entrance. The sensor will stay red color to indicate that there is no motion detected as shown in Figure 4.55. When there is motion detected, the PIR motion sensor will turn to green color as shown in Figure 4.56.



Figure 4.57 shows that the authorities received an SMS alert from Twilio when the PIR motion sensor detected motion in the laboratory.



Figure 4.57: Message Alert from Twilio

The problem of people forgetting to switch off all the appliances can be reduced by enabling the laboratory users to control and monitor the appliances even at a distance. From the website, the user can go to the dashboard and control the switches on-off as shown in Figure 4.58.



Figure 4.58: Lamp Switches

Figure 4.59 shows that both of the lamps lighten up once the user turns on the switches for LAMP1 and LAMP2.



Figure 4.59: Lamp Turned On

4.6 Discussion

This project can be concluded as successful putting as the monitoring systems are all functioning well. However, there are still some spaces for improvement.

First of all, the analysis should be added to the result to prove that this developed smart laboratory system can decrease the power assumption and save energy.

Second, the system can be improved if the automation technology is used to the fullest. The appliances in the library such as lights, fans, and air conditioners should have the ability to turn on automatically when there is motion detected and turn off automatically when there is no motion detected by the PIR motion sensor.

Lastly, the prototype presentation can be improved so that it looks more presentable. The first step is cutting the plywood with the appropriate size. Sandpaper is suggested to be used for a smoother plywood edge. A coat of polyurethane on the plywood surface and painting help to improve the prototype appearance. Labeling the name for all the components, wire connections, and laboratory areas in the prototype can make it more easy-understanding.

CONCLUSION

5

5.1 Conclusion

This study presented a cost-effective, user-friendly and reliable IoT-based smart laboratory system. A local server for the smart laboratory system has been successfully designed and built using Raspberry Pi, while the cloud server by using DigitalOcean. Besides, the communication between local/cloud servers and NodeMCU was established by using the MQTT protocol to display the real-time output. The real-time data monitoring and controlling using IoT technology helps to increase security and reduce energy consumption. The result of this study are promising and the developed system can function well with all the monitoring systems. The web page of the smart laboratory system is successfully developed for users convenient to access and view the real-time data of the laboratory anytime and anywhere.

5.2 Limitation

The developed smart laboratory system did not use the smart automation technology to the fullest as the appliances such as lights, fans, and air conditioners in the laboratory will not turn on or off automatically according to the motion detection. The laboratory users are only allowed to remotely control the appliances by accessing the website. It is great only when the users remember that they left the laboratory appliances on but it is not a good idea when the users do not remember that. It only helps to reduce the problem of high power consumption and energy wastage but did not completely solve the problem. Another limitation is the problem of the Wi-Fi connection. The system will stop functioning and the clients will not receive any data from the sensors once it stops detecting.

5.3 Recommendation

To overcome the limitation of this smart laboratory system, there are a few recommendations to improve in the future.

First, the appliances such as lights and air conditioners in the laboratory should be turned on automatically when there is motion detected and turned off automatically when there is no motion detected by the PIR motion sensor. A buzzer is suggested to be added to the component list. When the temperature is detected higher than the room temperature, a command will be sent to switch ON the air conditioner.

Secondly, RFID access cards could be added to the system to identify the users and trigger the relay module to open the entrance. For safety and security purposes, only the people who have predefined their RFID access card are allowed to enter the laboratory. Besides, an SMS alert can be sent to the authorities when detected a failed RFID is accessed up to 3 times. By receiving the alert notifications, the authorities can be aware that there are strangers or people who do not predefine their access card are trying to enter the laboratory. Actions can be taken rapidly if this kind of incident happens. A conveyor system helps to separate the components in the laboratory to prevent them from mixing. The knowledge of the Programmable Logic Controller (PLC) can be implemented in the next study to improve the systematic conveyor system that helps to separate the various types of components.

Next, a domain name can be purchased for the smart laboratory system website so that it can be easily accessed by users. Besides, the laboratory website should be limited to only members or students who can log in and get all the information instead of being open to everyone to increase security.

REFERENCES

- Amruta N. Banagar, and Rajshankar Khattar. 2020. "IoT Based Smart Laboratory System." International Journal of Engineering Research And V9(01):315–18. doi: 10.17577/ijertv9is010178.
- Ferencz, Katalin, and Domokos József. 2020. "Using Node-RED Platform in an Industrial Environment." (February):13.
- Jabbar, Waheb A., Senior Member, Tee Kok Kian, Roshahliza M. Ramli, Vladimir Shepelev, and Soltan Alharbi. 2018. "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System." *IEEE Access* XX.
- Jhang, Wei Hong, Liang Bi Chen, Wan Jung Chang, Che Ching Yang, and Chao Tang Yu. 2017. "Design of a Low-Cost Level-Triggered Zigbee Network Multi-Application Sensor in Smart Homes." 2017 6th International Symposium on Next Generation Electronics, ISNE 2017. doi: 10.1109/ISNE.2017.7968729.
- Longe, O. M., K. Ouahada, S. Rimer, H. Zhu, and H. C. Ferreira. 2015. "Effective Energy Consumption Scheduling in Smart Homes." *IEEE AFRICON Conference* 2015-Novem. doi: 10.1109/AFRCON.2015.7331917.
- Longo, Edoardo, Alessandro E. C. Redondi, Matteo Cesana, Andres Arcia-Moret, and Pietro Manzoni. 2020. "MQTT-ST: A Spanning Tree Protocol for Distributed MQTT Brokers." *IEEE International Conference on Communications* 2020-June. doi: 10.1109/ICC40277.2020.9149046.
- M. Elatawy, Safaa. 2020. "IoT-Based Smart Lab System in Schools Using Arduino and Bluetooth Based Android Smartphone." *International Journal of Computer Applications* 175(19):52–59. doi: 10.5120/ijca2020920724.
- Patterson, Murray G. 1996. "What Is Energy Efficiency? Concepts, Indicators and Methodological Issues." *Energy Policy* 24(5):377–90. doi: 10.1016/0301-4215(96)00017-1.
- Poongothai, M., and P. Muthu Subramanian. 2018. "Design and Implementation of IoT Based Smart Laboratory." *Proceedings of 2018 IEEE Applied Signal Processing Conference*, *ASPCON 2018* 19–23. doi: 10.1109/ASPCON.2018.8748696.
- Ruwaida, , Bassam, and Toni Minkkinen. 2013. "Home Automation System : A Cheap and Open-Source Alternative to Control Household Appliances."
- Ryan, J. L. 1988. "Home Automation." IEE Review 34(9):355-58. doi: 10.1049/ir:19880143.
- Sanduleac, Mihai, Constantin Bulac, Vladimir Tanasiev, Dorel Stanescu, Carmen Stanescu, and Monica Florea. 2017. "Energy Awareness, an Important Goal for Empowering the End Customer." Proceedings - 2017 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2017 and 2017 Intl Aegean Conference on Electrical Machines and Power Electronics, ACEMP 2017 599–604. doi: 10.1109/OPTIM.2017.7975034.

Sriskanthan, Nadarajah, Forest Tan, and Advait Karande. 2002. "Bluetooth Based Home Automation System." *Microprocessors and Microsystems* 26(6):281–89. doi: 10.1016/S0141-9331(02)00039-X. APPENDICES

Project Activities	2021	2021	2021	2021	2021	2021	2021	2022	2022
	JUN	JUL	AUG	SEP	ОСТ	NOV	DEC	JAN	FEB
Introduction									
Literature review study									
Methodology									
Product Fabrication									
Project Outcome									
Technical Report									
Presentation									
Thesis writing									
Final work submission									

Appendix A: Gantt Chart for Senior Project

Appendix B: PHP Coding



×1 F		Selection View Go Run Terminal Help index.php - Webpage - Visual Studio Code			
Ch	🖙 index	.php ×			
		> 🕸 index.php			
ρ		<pre><div class="icons"></div></pre>	102		
ŕ		<pre><i class="fas fa-search" id="search-btn"></i></pre>			
90		<i class="fas fa-user" id="login-btn"></i>			
8			Dare Dare		
~					
a^		<form action="" class="search-bar-container"></form>			
		<input id="search-bar" placeholder="search here" type="search"/>			
EP-		<pre><label class="fas fa-search" for="search-bar"></label> </pre>	Ĵ	ITTEL	
			38		
		al handen portion and a	 		
		<pre><!-- login form container--></pre>			
			2	kig	
		<pre><div class="login-form-container"></div></pre>	3	inin	
		<i class="fas fa-times" id="form-close"></i>			
				AND LOTTER	
		<form action=""></form>		Street.	
		<h3>login</h3>		85	
		<input class="box" placeholder="enter your email" type="email"/>			
		<input class="box" placeholder="enter your password" type="password"/>			
		<input class="btn" type="submit" value="login now"/>			
		<pre><input la_="" remember="" type="checkbox"/> </pre>			
		(label for= remember >remember me			
\bigcirc		<pre></pre> cp>torget password: clitck interex/a>			
V (1)		//forms			
~~					
500					
Ø 0 /	A 0	Ln 68. Col 1. Spaces 4. UT	F-8 CRLF	PHP	R (1











×1 F	ile Edit	Selection View Go Run Terminal Help index.php - Webp	ige - Visual Studio Code	- 0	×
Ch	🐄 index	x.php ×			
	new try	/ > 🐄 index.php			
ر ھ		<pre>cp>TG18040 </pre>			
₽					-
₿					
		footer section			
		<section class="footer"></section>		7885	
		<div class="box-container"></div>			
		<pre><div class="box"></div></pre>			
		<pre><i class="fab fa-instagram"> Instagram</i></pre>		Janser-	
		<pre><i class="fab fa-whatsapp"> Whatsapp</i></pre>		- Market	
				125	
0		(script spc="https://uppkg.com/swiper/swiper-bundle min_is"\//script\			
V (1)		- (Seript Ste- netps://unpkg.com/swiper/swiper-bundle.min.js //seript/			
~~					
500		<pre><script src="newjavascript.js"></script></pre>			
⊗ 0 Z	<u>\</u> 0		Ln 61, Col 12 Spaces: 4 UTF-8 (CRLF PHP 🔗	C



Appendix C : CSS Coding

×1 -	File Edit	Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code	– 0 X
Сh	# newc	33.035 X	
	new try	> # newcs.css >	
0		<pre>@import url('https://fonts.googleapis.com/css2?family=Verdana:wght@200;300;400;600;700&display=swap');</pre>	The same is an a set of the second set of the second set of the second s
\sim			All the second s
0.0		:root{	
R		darkcyan: 🗖 darkcyan;	Bon-
			1928
3			Without .
-0		font-family: 'Verdana', sans-serif;	
6		margin:0; padding:0;	
		DOX-SIZINE: DOPOR-DOX;	
		outline: none;	WHICH .
		taxt.deconstion: none:	
		transition: all .25 linear:	18t.
			in the second seco
		*::selection{	Sector of the se
		background:var(darkcyan);	
		color: ##fff;	
			BALLER TO A
		html(BURNER BURNER BURNER
		font-size: 55%;	
		align-items: center;	100 m
		overluw-x: nidden;	
		scroll-padoulng-top: orem;	Mariana .
		3	The second second
0			North Contraction
<u></u>		section{	
		padding:2rem 9%;	Wer .
567			
	33		
⊗0.	A 0	Ln 1, Col 1 Spaces: 2 U	JIF-8 CRLF CSS 📯 🖵

	File Edit Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code	– 0 ×
β	# newcss.css ×	□ …
	new try > # newcss.css >	
ھ ھ	34 .heading{ 35 text-align: center; 36 text-transform: uppercase; 37 text-shadow: ■lightgrey 2px 2px;	
₽ ₽	<pre>38 font-weight: bolder; 39 font-size: xx-large; 40 padding: 2.5rem 0; 41 letter-spacing: 3px;</pre>	1000 1000 1000 1000- 1000- 1000- 1000- 1000- 1000- 1000- 1000- 1000
Θ	42 color: □darksyan; 43 } 44 45 button{ 46 display: inline-block; 47 margin-top: irem;	in a second seco
	48 background: ■darkcyan; 49 color: ■#fff; 50 padding:.8rem 3rem; 51 border:.2rem solid = white; 52 cursor: pointer; 53 font-size: 1.7rem;	Bit in the second secon
	34 [TOTE-Weign:: bold; 55 } 56 57 button:hover{ 58 background: ■white; 59 color: ■darkcyan; 68 border:.2rem solid ■darkcyan;	195° 1957 1967 1967 1967 1967 1967 1967 1977 197
€	01 } 62 63 button:active { 64 box-shadow: 2px 5px ■#666; 65 transform: translateV(4px); ▲0	In 1, Col 1 Spaces: 2 UTF-8 CRLF CSS & C

×1	File Edit	Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code		
Ch	# new	KSS.CSS X		
	new try	y > # newcss.css >		
Q				Bar-
1		button:active {		inte-
90		box-shadow: 2px 5px 🔲 #666;		and the second s
8		<pre>transform: translateY(4px);</pre>		Winn.
~				10
a^				
		header4		The second se
EP-		position: Tixed;		
		background: <u>H</u> #33:		Saturies
		z-index: 1000:		BOS.
		display: flex;		
		align-items: center;		ageneration of the second seco
		justify-content: space-between;		Bigen .
		padding: 0.5rem 5%;		100 mm
				020 gene
		header .logoi		BELOW
		with seeks;		Training and the second
		t cursor.pointer;		jane West
				855-
		header .navbar af		
		color: #fff;		The second secon
		font-size: 2rem;		
		margin:0 .8rem;		
				Mar
0				BOW Name and the
K		neader navoar allover{		
-04		Cotor: Budrkeyan;		Barran an
Ê				MAC
⊗ 0	∆0		Ln 1, Col 1 Spaces: 2 UT	F-8 CRLF CSS 🕅 🗳



×1	File Edit	Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code			σ×
Сh	# newo	ssees ×			
		> # newcss.css >			
Q				1139%	
/-		header .search-bar-container #search-bar{		185-	
90		width:100%:			
82		padding:1rem;			
		text-transform: none;		The second	
~~		color:□#333;			
~		font-size: 1.7rem;			
<u>_</u>		border-radius: .5rem;		Box.	
				38E	
				Barbarban Aggerene en Aggerene en	
		header .search-bar-container label{		1000 m	
		color: #fff;		192	
		cursor: pointer;		Sector of the se	
		font-size: 3rem;			
		margin-left: 1.5rem;		BEEX.	
				ST.S.	
		hadan search-har-containen lahal-hoven/		Was-	
		colorisation adversal.		195-	
] - COLOR (COLOR (COLOR (
				TELEVISION IN CONTRACTOR	
		.login-form-container{			
		position: fixed;			
		top:-120%; left: 0;		Man	
		z-index: 10000;			
		min-height: 100vh;		20.05	
		width:100%;		1925/C	
0		background: 🗆 rgba(0,0,0,.7);		3040/00 ····	
-64		display: flex;		NEW STEP AND	
		align-items: center;		202	
563		justify-content: center;		88	
	151	}		1997	
(⊗ n	<u>A 0</u>		1 Col 1 Spaces: 2 UTE-	8 CRLF CSS	8 C

×1	File Edit Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code	- 0 X
ſЪ	T # newcs.css X	
	new try > # newcss.css >	
	153 .login-form-container.active{	
	154 top:0;	
	150 login-form-container form	The ADMINISTER AND ADMINISTER AND ADMINISTER
.ġ>	158 margin:2rem;	BDC.
	159 padding:1.5rem 2rem;	No.
RP	160 border-radius: .5rem;	Berner an Berner an
	161 background: ##fff;	Bartan Bartan
	162 width:50rem;	Misson
		NUMBER OF A DESCRIPTION OF A DESCRIPTION OF A DESCRIPTION
	104 login-form-container form h3{	Takara
	160 font-size: 3rem;	
	167 color: 🗆 #444;	B.C.
	168 text-transform: uppercase;	
	169 text-align: center;	3.47-
	170 padding:1rem 0;	No.
		Hit
	1/2 173 login-form-container form how	
	175 width:100%	North Contraction of
	175 padding:1rem;	Bir
	176 font-size: 1.7rem;	1965
	177 color:□#333;	Water and Street
	178 margin:.6rem 0;	West
	179 border: 1rem solid Lingba(0,0,0,.3);	MACTINE AND A
8	180 Cext-transform: none;	
		18.8 19.00
563	183 .login-form-container form .box:focus{	and the second s
	184 border-color: var(darkcyan);;	
0	184 border-color: var(darkcvan);	

×1 -	File Edit Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code		- 6	
C)	# newcss.css ×			
-	new try > # newcss.css >			
Q	182 163 .login-form-container form .box:focus{ 184 border-color: var(darkcvan)::		ju vite- ju vite- ju o an	
ge			Bartantierte a se	
æ	18/ .login-form-container form #remember{ 188 margin:2rem 0; 190 l			
₿	199 190 191 .login-form-container form label{			
	192 font-size: 1.5rem; 193 }		Section 2019	
	194 195 login-form-container form .btn{ 196 display: block:		Ner-	
	197 width:100%; 198 }		No. Maria	
	199 .login-form-container form p{ 201 padding:.5rem 0; 202 font-size: 1.5rem;		Millionan Millionan Million Million Million	
	203 color: ₩#666; 204 } 205		NATION CONTRACTOR	
	206 .login-form-container form p a{ 207 color:var(darkcyan);			
	206 } 209 314 login-form-container form n ::boyer(No.	
8	211 color: U#333; 212 text-decoration: underline;			
501	213 }		accura	5 (
∞ 0⊿		En I, COLL Spaces: 2 UTF-8	CREF CSS	8 4



×1 -		Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code			
Ch	# newcs	x x			
		# newcss.css >			
Q					
1		.home .content h1{			
90		color: ■#fff;			
8		text-transform: uppercase;		All and a second s	
~		text-shadow: 0 .3rem .5rem 🗆 rgba(0,0,0,.5);			
		font-size: 30px;			
				Sectored States	
H-P-P				BUSE	
		home content has			
		color: ####			
		text-transform: lowercase;			
		font-size: 15nv:			
		font-weight: lighter:			
		}		Martine .	
				Ser -	
		.home .content p{		Man	
		font-size: 2.5rem;			
		color: ##fff;			
		padding:.5rem 0;		1627.	
		text-transform: lowercase;			
		text-shadow: 0 .3rem .5rem □rgba(0,0,0,.1);		BOAL STREAM	
	268			POL-	
		home wides containen wides		Sec	
		nome video-container videot			
		top:0: left: 0:		All Vitte or non or All source over the	
_		z-index: -1;			
8		height: 100%;		180	
		width:100%;			
523		object-fit: cover;		BM9-	
- MI		}			
Ø 0.	A 0		In 1 Col 1 Spacer 2		

×	📢 File Edit Selection View Go Run Terminal Help new	ss.css - Webpage - Visual Studio Code	-	٥	×
ſ'n	m # newcss.css ×				
	new try > # newcss.css >				
0	Q 278				
	279 .home .controls{				
	280 padding:1rem;		Bigan .		
	281 border-radius: srem; 282 background: praba(0.0.0.7):		ENC.		
	283 position: relative:		Second Second		
±\$	284 top: 10rem;				
			anner in see		
ß	286		E.S.		
	287 .home .controls .vid-btn{				
	288 height:2rem;				
	289 Width:2rem;				
	290 border-radius: 50%:		The second		
	292 background: #fff;		Service and a		
	293 cursor: pointer;		Mar-		
	294 margin:0 .5rem;		Ke-		
			Mar.		
			BEE		
	297 .home .controls .vid-btn.active{		States and		
	298 Dackground: Darkcyan;		BOG.		
	300		965		
	301 .services .box-container{		18.92. 18.92.		
	302 display: flex;		Ke		
	303 flex-wrap: wrap;				
	304 gap: 1.8rem;				
			160		
Ø	Q 306		NO.		
U	307 .services .box-container .box{		3.85.		
	308 TIEX: 1 1 30/Pem;		Marrie Marrie		
20	310 nadding: 3rem 2rem:		86851		
⊗0.		Ln 1, Col 1 Spaces: 2 UTF-8	CRLF CS	s <i>ন</i> ি	4

	File Edit Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code		b ×
C)	# newss.cs ×		
	new try > # newcss.css >		
く よ の 品	<pre>let inty / w intercess / 306 307services .box-container .box{ 308 filex: 1 1 30ren; 309 border-radius: 1.5ren; 310 padding: 3ren zren; 311 text-align: center; 312 background-color: @darkcyan; 313 } 314 315 .services .box-container .box i{ 316 padding:1ren; 317 font-size: Sren; 318 color: @wheet; 319 text-shadow: 4px 2px @darkgray; 320 }</pre>		
6	<pre>services .box-container .box h3{ color: ■wheat; color: ■wheat; color: ■wheat; color: ■wheat; color: ■white; padding: 2rem 0; color::■white; color::=white; color::■white; color::</pre>		
500	337 } 338	Martin Carlos Carlos	

×1	File Edit	Selection	View Go Run	Terminal	Help		/ebpage - Visu	al Studio Code					-	٥	×
Сh	# newo	css.css ×													
-	new try	/ > # newcs:													
ر م		.gallery display flex-w	.box-containe : flex; pap: wrap;	in{											
م ح		gap: 1. }	5rem;										BELOW TO A		
&∕ ⊓		.gallery overflo	.box-containe w: hidden;	er .box{									1965- 1965-		
Ш		box-sha border: border-	dow: 0 1rem 2 1rem solid radius: .5rem	rem ⊡rgb a ∎#fff; I;	a(0,0,0,.5);								internet		
		flex:1 height:	1 30rem; 30rem;												
		}											9607 9607		
		.gallery height: width:	.box-containe 100%; 100%;	r .box img	3{								NACES IN COLUMN STREET		
		object }											9.87.77.7 9.87.77.7		
		.gallery	.box-containe m: absolute;	r .box .co	ontent{								967 967 967		
		top:-10 height: width:	00%; left:0; 100%; 00%:										1990 1962 343		
8		text-al backgro	lign: center; pund:□rgba(0,	0,0,.45);									New Constant		
£20	369 370	padding padding }	; 2rem; ;-top: 10rem;										Martin Carlos and Anna anna a		
⊗ 0	Δ0									Ln 1, Col 1	Spaces: 2	UTF-8	CRLF CS	s 8	C

×1 F	File Edit Selection View Go Run Terminal Help newss.css - Webpage - Visual Studio Code	
Ch	# newcss.css ×	
	new try > # newcss.css >	
Q	372	
<i>′</i>	373 [top:0;	TRACE
e٩		COUNTRY OF
0		BAC" Block
A	376 .gallery .box-container .box .content h3{	195-
æ	3/7 color: wheat:	
-0		
Ш		Birm.
	381 .gallery .box-container .box .content p{	Er
	382 font-size: 1.5rem;	
	383 COLOT: Trees	
	304 pauling. Trem 0, 285 text-transform: Joweprace:	No. of the second
		NEW COLOR AND A
	388 .about.about-slider{	The second se
	389 padding-bottom: 2rem;	128." 128."
	392 .a00UT.b0X{	
	Job pauling. Iren; 304 text-align: conter:	100 mm
	395 box-shadow: 0 irem 2rem □rgba(0,0,0,.5);	BLC. EXco-
	396 border-radius: 5px;	2020-20 2020-20 2020-20
	397 background-color: lightgrey ;	Billion
	398 height: 40vh;	NGN -
0		No. of State and Stat
V 1	400	
< <u>7</u> 2	402 beidetiscon ang	
200	403 with:15rem;	Jan and Anna
⊗ 0 ∠	∆0 Ln1, Col 1 Spaces:	2 UTF-8 CRLF CSS 🔗 🤩



×	File Edit	Selection View Go Run Terminal Help newcss.css - Webpage - Visual Studio Code	í i	– 0 ×
Ch	# new	333.633 X		
	new try	> # newcss.css >		
0				
/-		.footer .box-container .box h3{		
00		tont-size: 2.5Pem;		
8		color: stem 6;		
		3		
_ a >				Berger -
		.footer .box-container .box a{		
		font-size: 2rem;		No.
ш		padding-right: 10px;		
		color:∎rgb(255, 255, 255);		
				We have a second
				MALL STREET, S
		.TOOTEP .DOX-CONTAINEP .DOX aINOVER		JAN T
	444	color: Dequestion: dimerine,		
)		
		.footer .box-container .box a i{		
		font-size: 15px;		100
		color: Egrey;		
		transition: transform 0.1s;		1997- 733.
		gap: 20px;		
				3657
		fonter hav-container hav achover if		The second se
		transform: scale(1.1):		
8				
		.footer .box-container .box a:hover .fa-facebook{		
		color: 🗐 #3b5998;		
-Em				Karmer.
00	462		In 1 Col 1 Second 2	







×1 -		Selection View Go Run Terminal Help ne	wcss.css - Webpage - Visual Studio Code	
ſЪ	# newcs	.css ×		
	new try	# newcss.css >		
0				
\sim		header .navbar.active{		Bill Without a
0-		clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);		84_ 1550-
R.				No.
		neader .navbar a{		
~		border-radius: Srem:		Bill turns
		padding:1.5rem:		NOT THE REPORT OF
ш		margin:1.5rem 0;		
		background: #222;		an Area and
		text-transform: uppercase;		
				Set over a set of the
				Marin Marine
		@media (max-width:450px){		ENG-
		htm]/		
		font-size: 50%		
				BARDA
				MACTERS on a state
		.heading span{		
		<pre>font-size: 2.5rem;</pre>		
		.contact .row form .inputBox input{		
		width:100%;		The Party of the P
8	545			1862 Jun
502				
-				
⊗ 0 .	∆ 0		Ln 1, Co	bl 1 Spaces: 2 UTF-8 CRLF CSS 🔊 🗘
Appendix D : JavaScript Coding



