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 THESIS 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 classified as:

☒	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☒	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☒	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. 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	Name of Supervisor
Date: 15 January 2022	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 real-time 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

LIST OF TABLES

LIST OF FIGURES

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
HAP	Home Automation Protocol
FTP	File Transfer Protocol
SFTP	Secure File Transfer Protocol

LIST OF APPENDICES

1

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

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 cost-effective 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.

2

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

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.

3

METHODOLOGY

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.

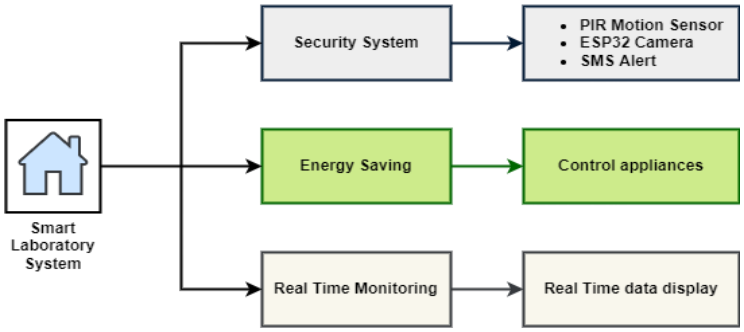


Figure 3.1: Smart Laboratory System

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.

Table 1: List of Components

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.

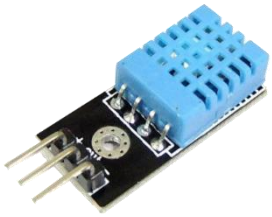


Figure 3.5: DHT11 Temperature and Humidity Sensor

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 entrance for motion detection. The ESP32-CAM is placed in front of the laboratory to display a real-time full eye-view vision of the laboratory. The lightbulbs are also placed in front of the laboratory.

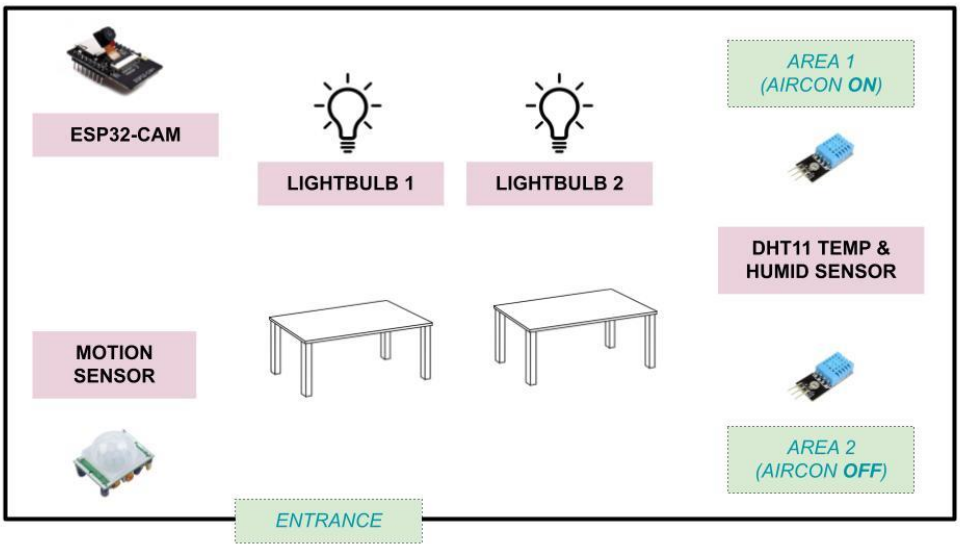


Figure 3.9: Smart Laboratory Layout

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.

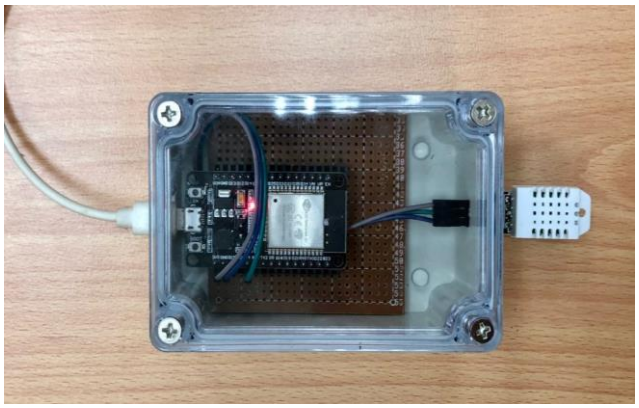


Figure 3.10: DHT11 Temperature and Humidity Sensor Hardware

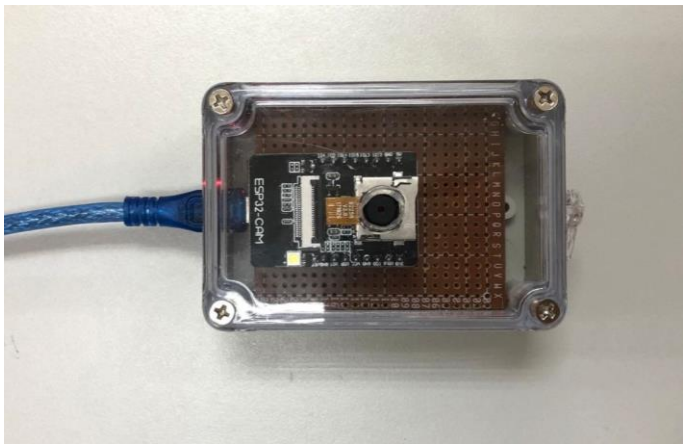


Figure 3.11: ESP32-CAM Hardware

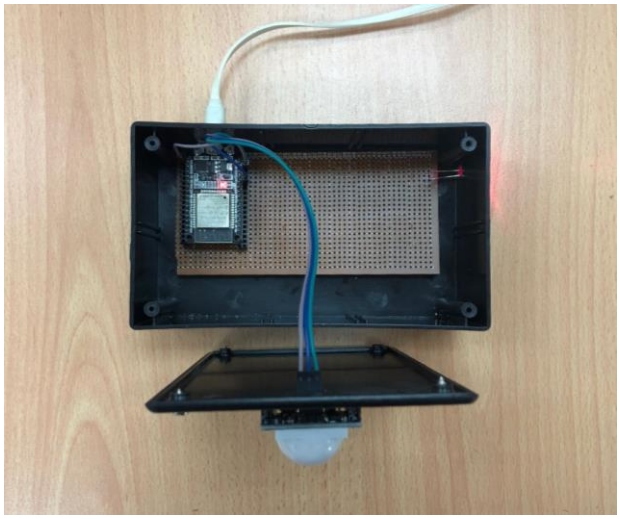


Figure 3.12: PIR Motion Sensor Hardware

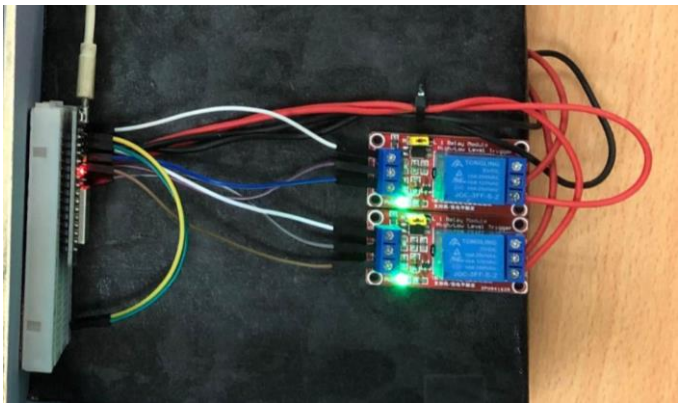


Figure 3.13: Relay Module Hardware

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.

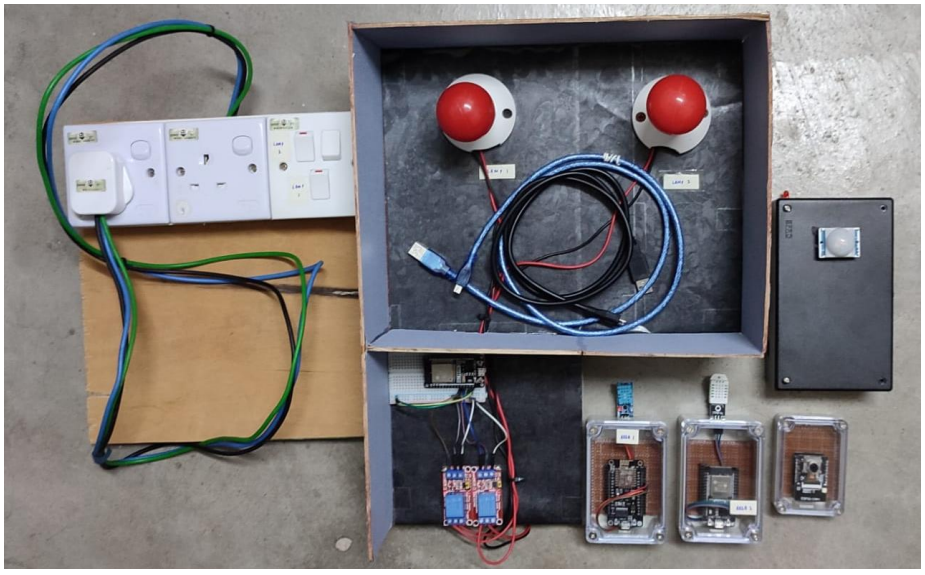


Figure 3.15: Prototype of Smart Laboratory System

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.

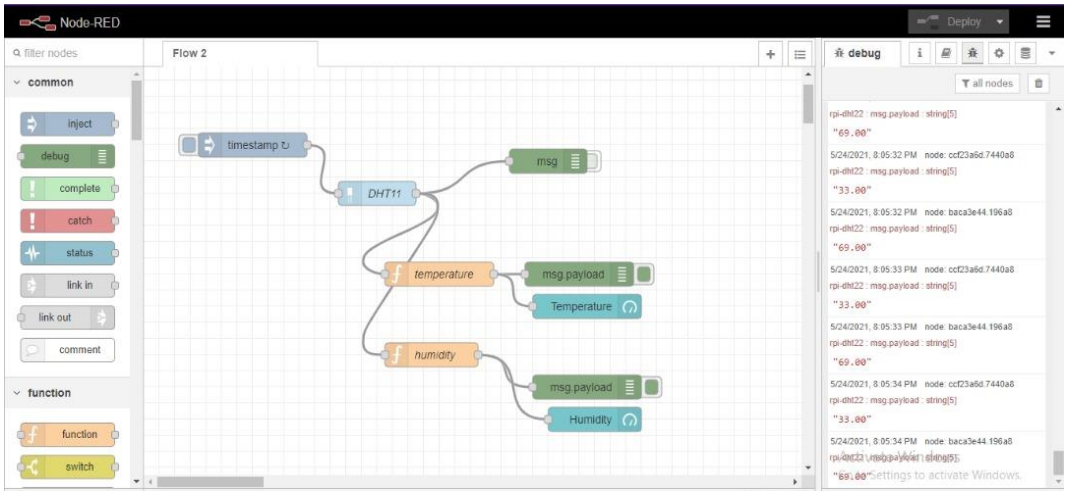


Figure 3.16: Node-RED programming for DHT11 Sensor

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.

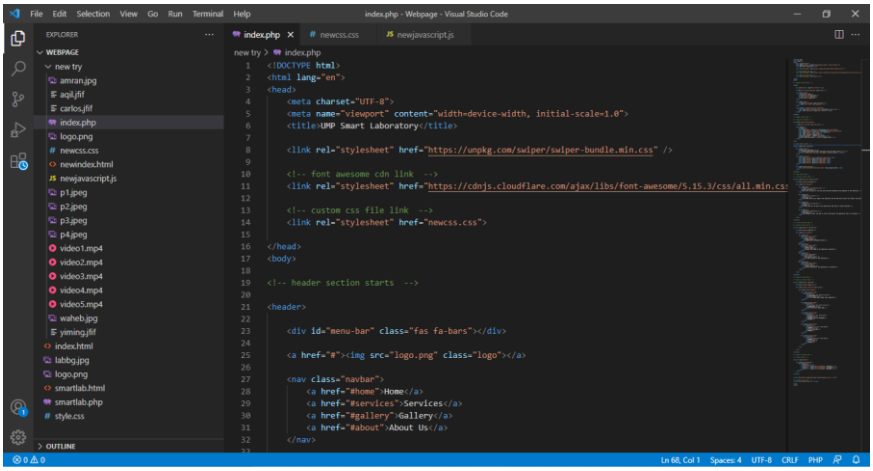


Figure 3.18: Visual Studio Code

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.

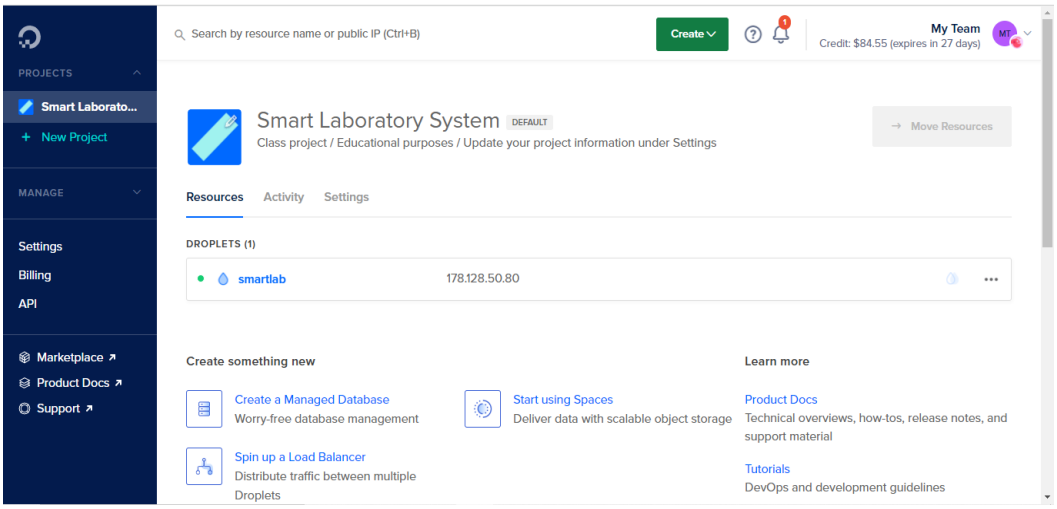


Figure 3.19: Digital Ocean

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.

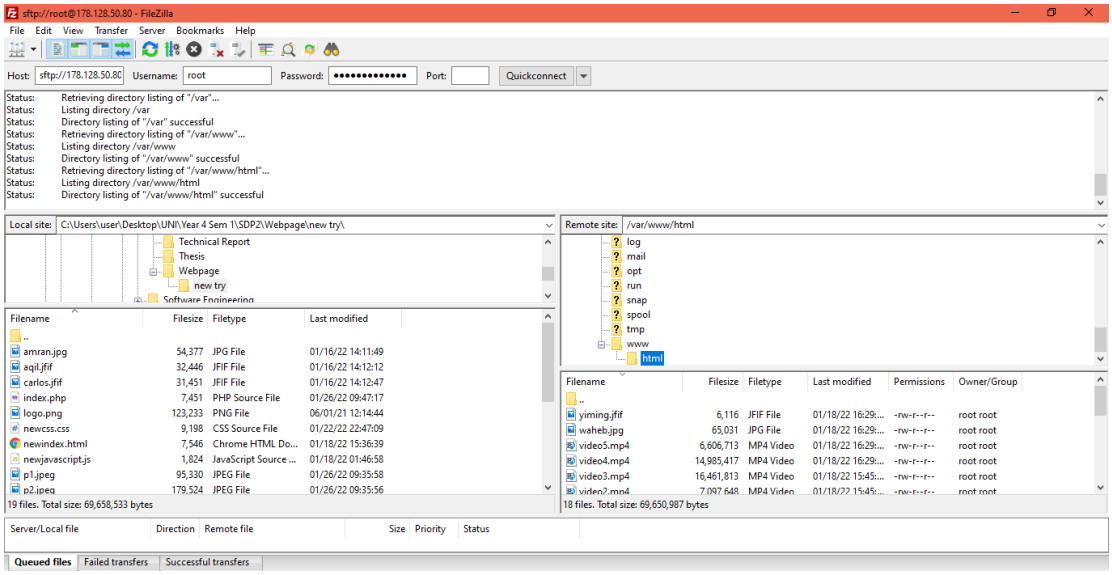


Figure 3.20: FileZilla

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.

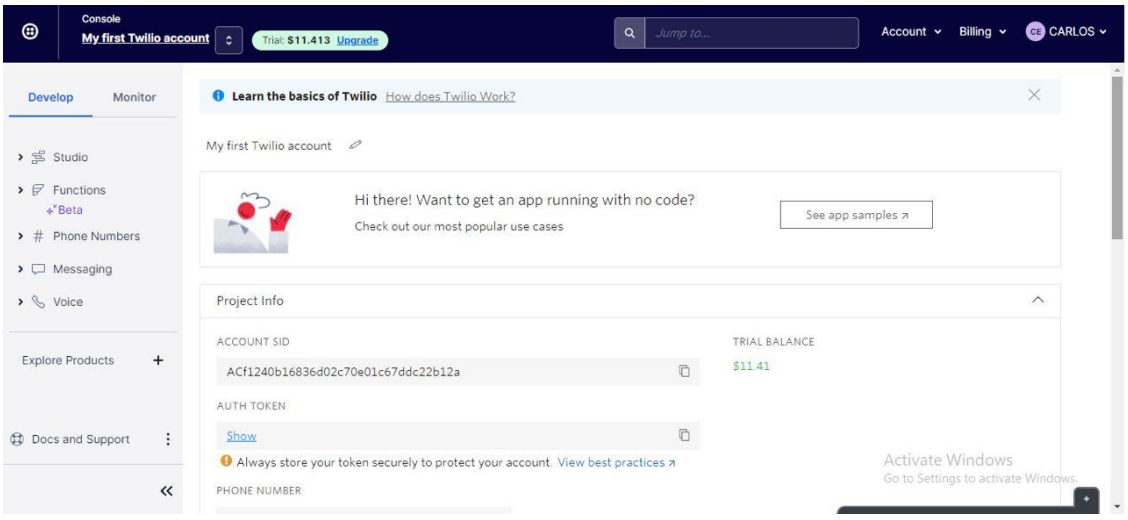


Figure 3.21: Twilio

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

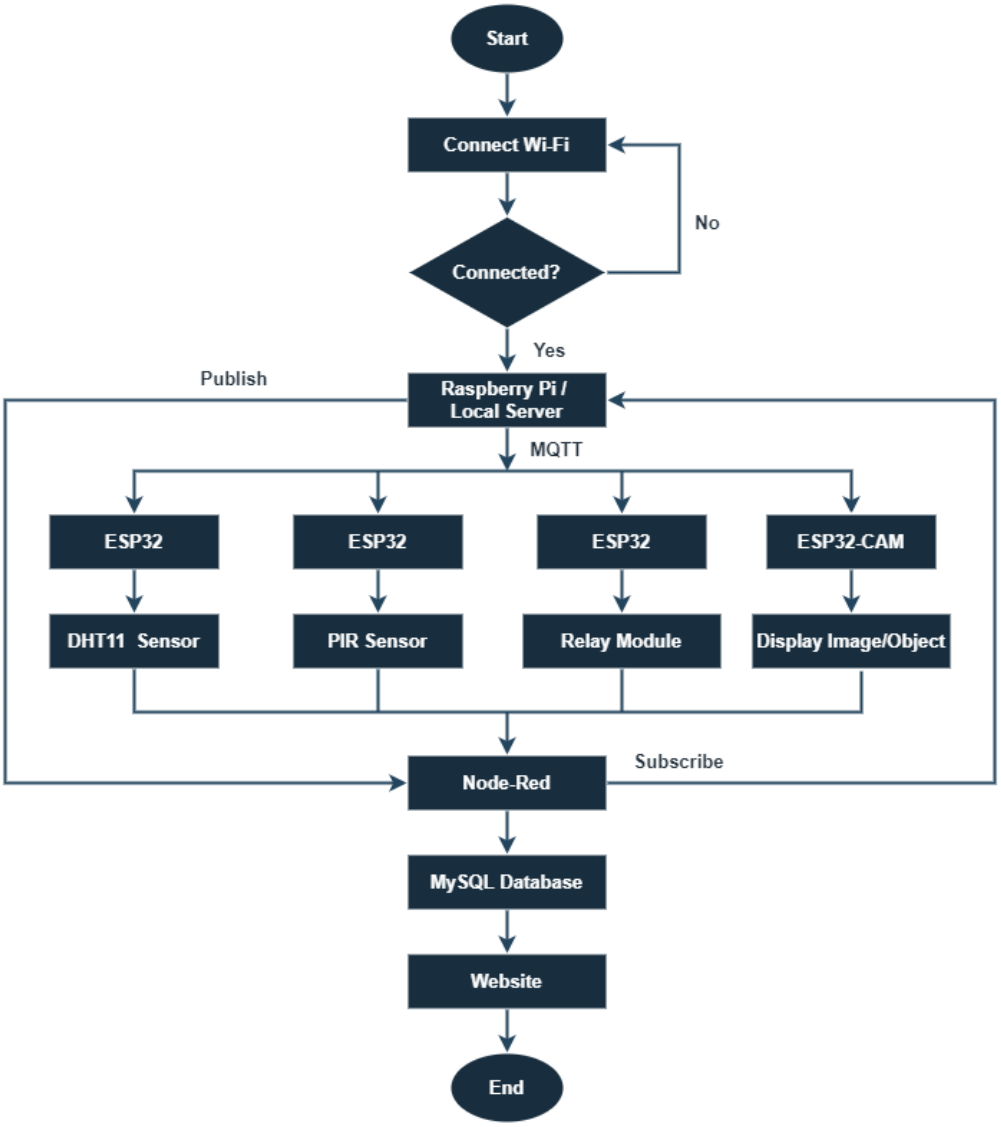


Figure 3.23: System Flowchart

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.

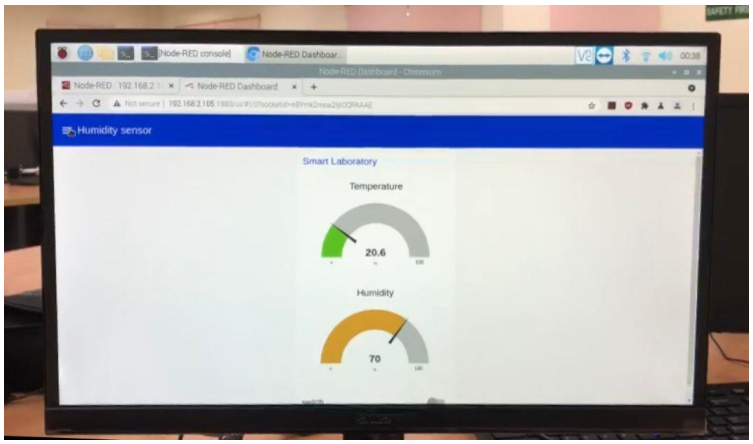


Figure 3.24: Testing of DHT11 Sensor with Output Readings

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.

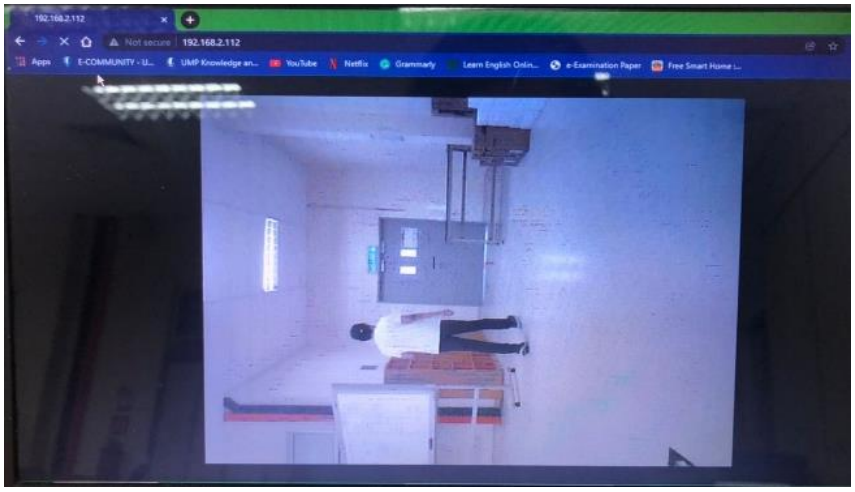


Figure 3.26: Testing of ESP-CAM Display

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.

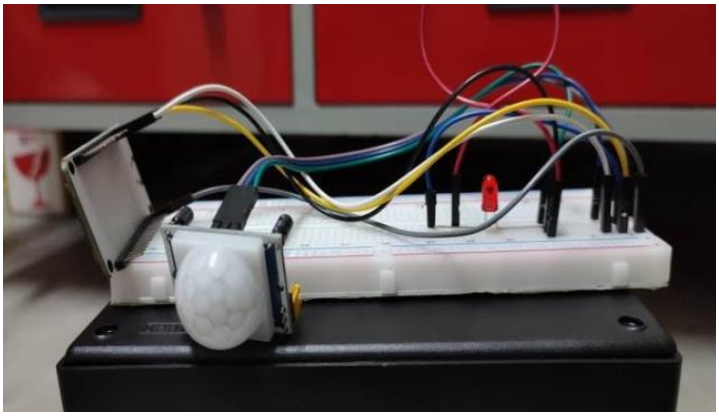


Figure 3.27: Circuit connection of PIR motion sensor

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.

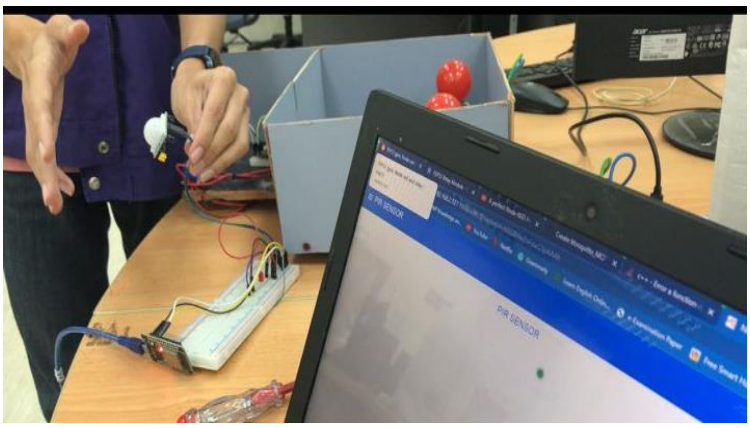


Figure 3.28: Testing PIR Motion Sensor in Short Distance

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.

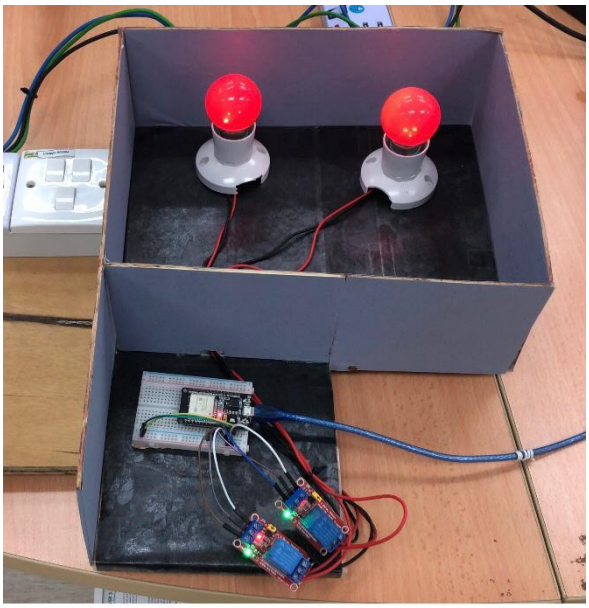


Figure 3.30: Circuit Connection of Relay Modules and Light Bulbs

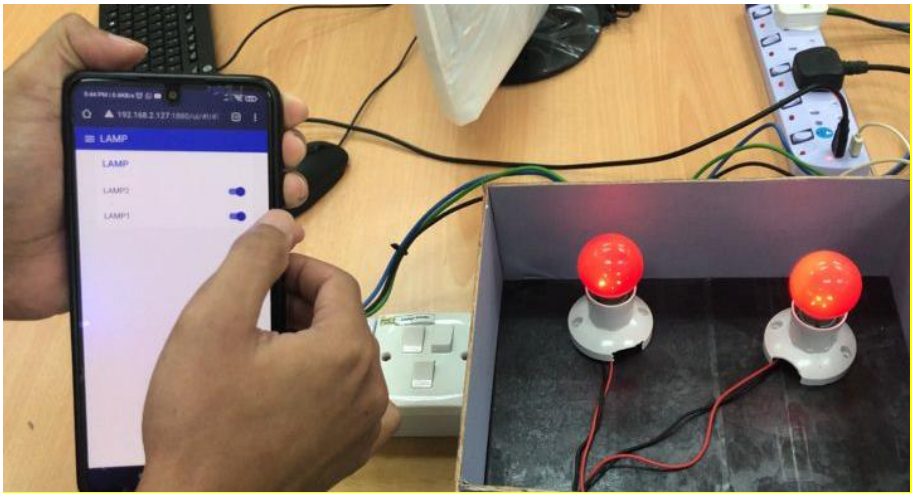


Figure 3.31: Testing for Controlling Light Bulbs

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.

Table 2: Cost Analysis

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 Humidity Sensor	2	RM 2.90	RM 5.80
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
Total Cost				**RM 352.25**

4

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 https://www.digitalocean.com as shown in Figure 4.1.

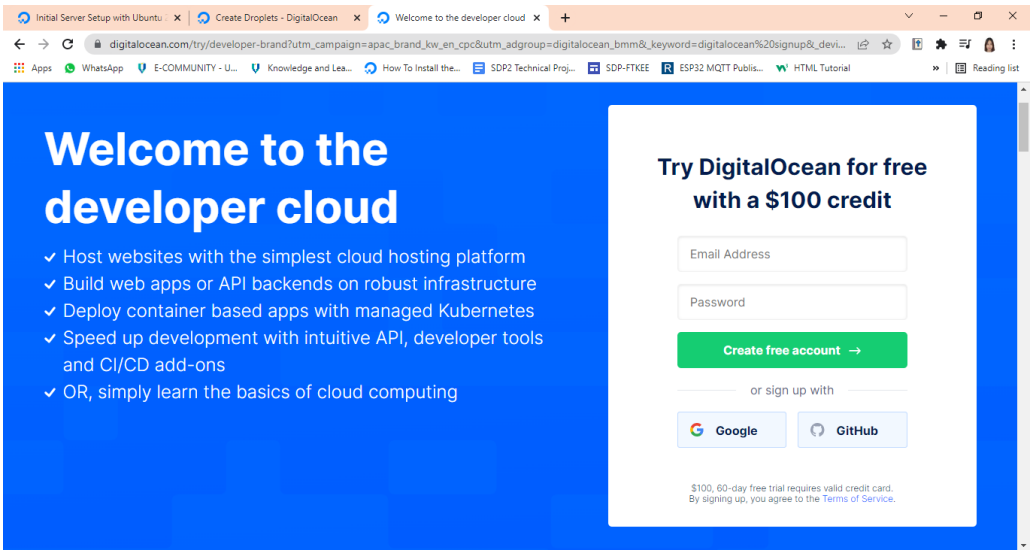


Figure 4.1: DigitalOcean Sign Up

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.

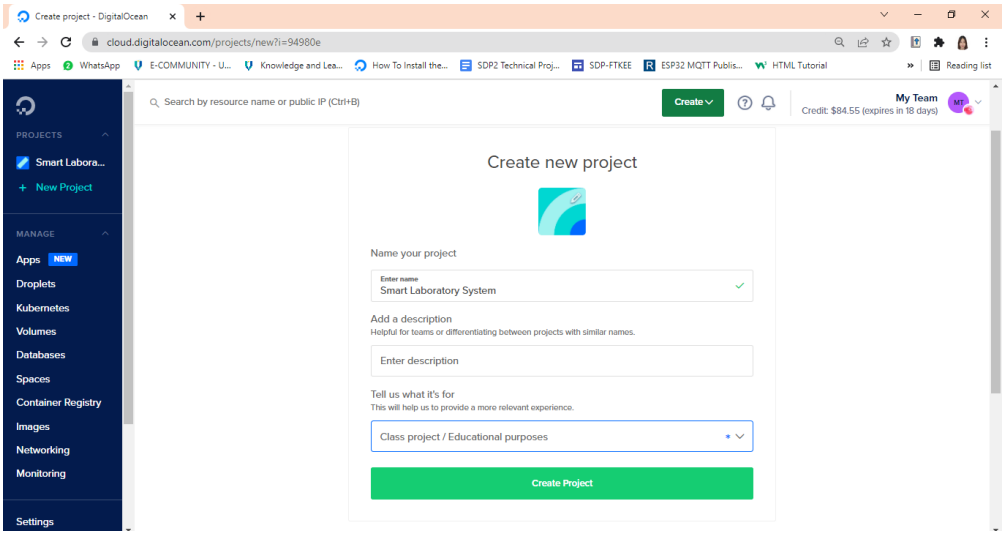


Figure 4.2: Create New Project at DigitalOcean

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.

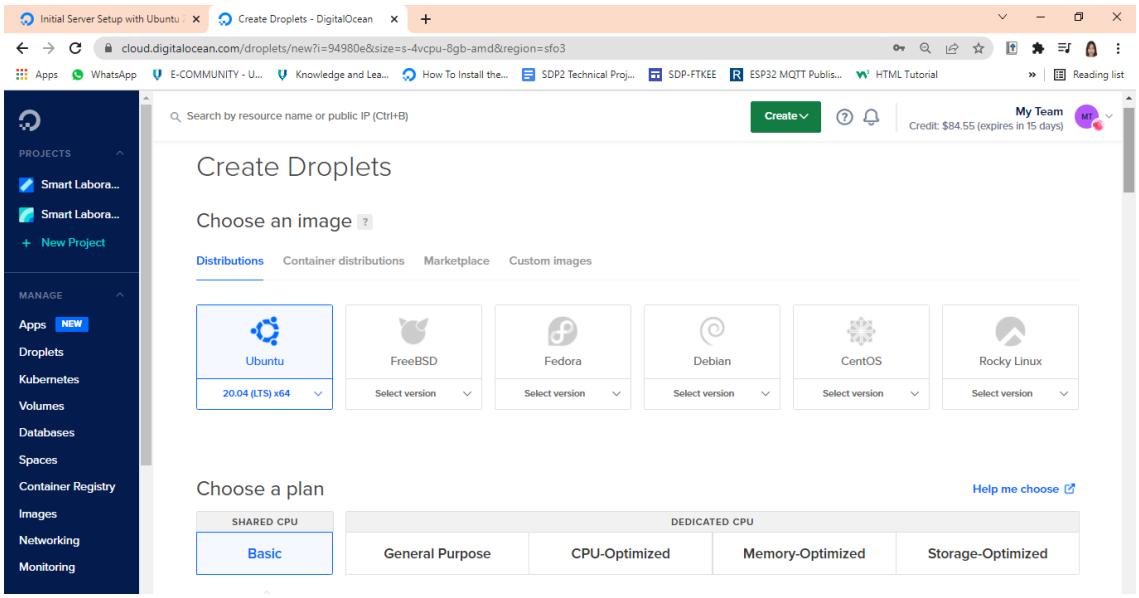


Figure 4.3: Image and Plan

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).

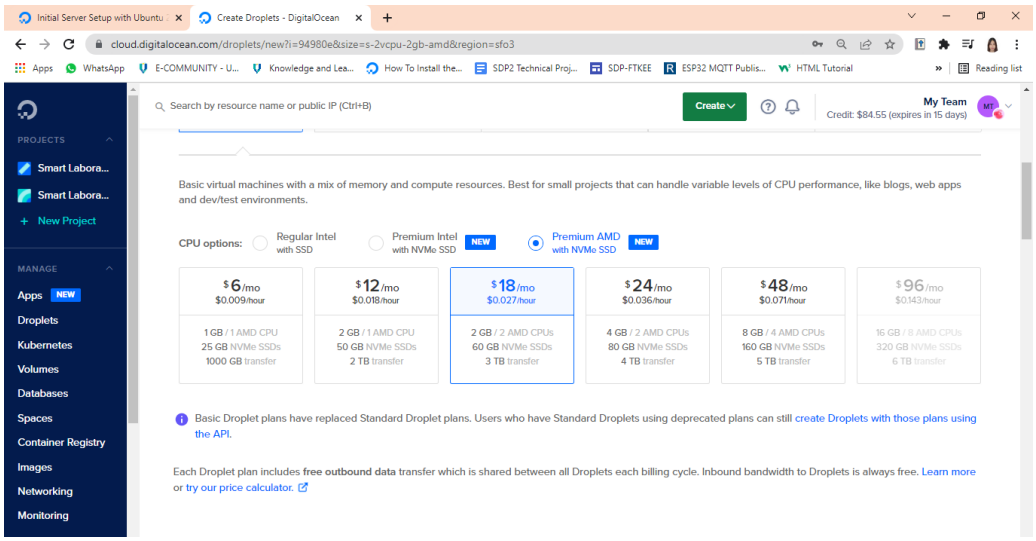


Figure 4.4: CPU Options

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.

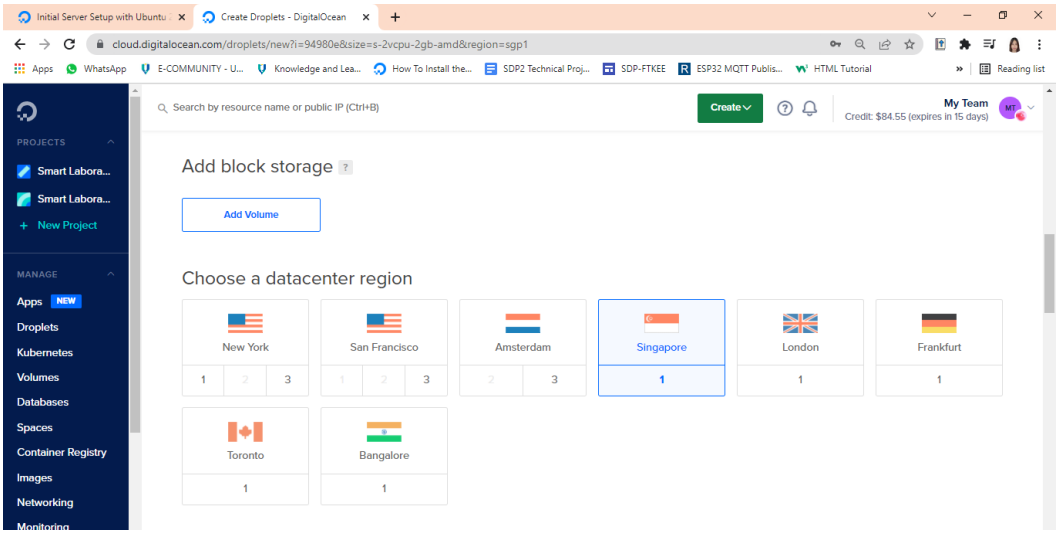


Figure 4.5: Datacenter Region

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.

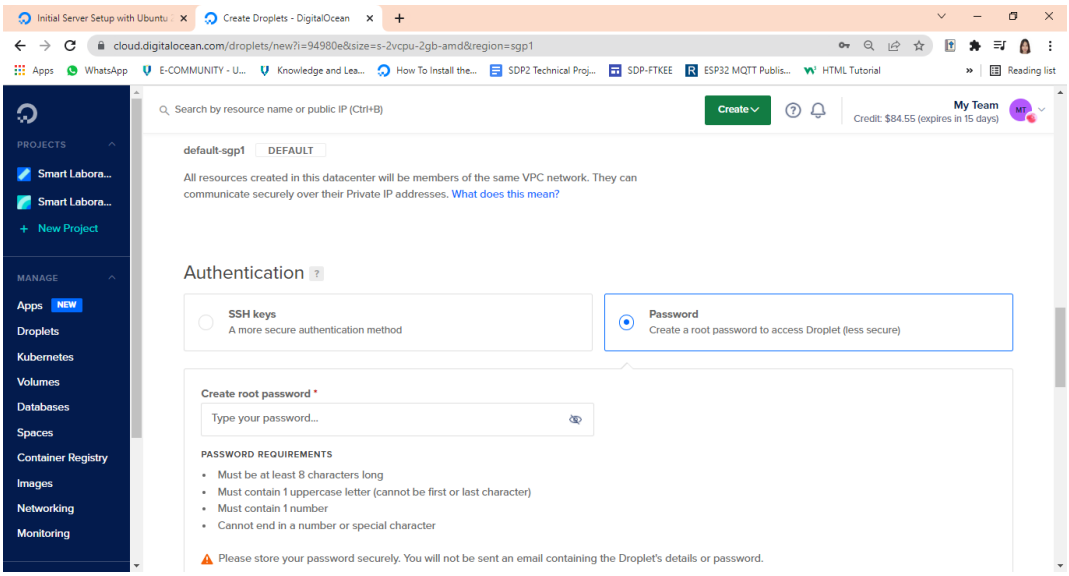


Figure 4.6: Authentication

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

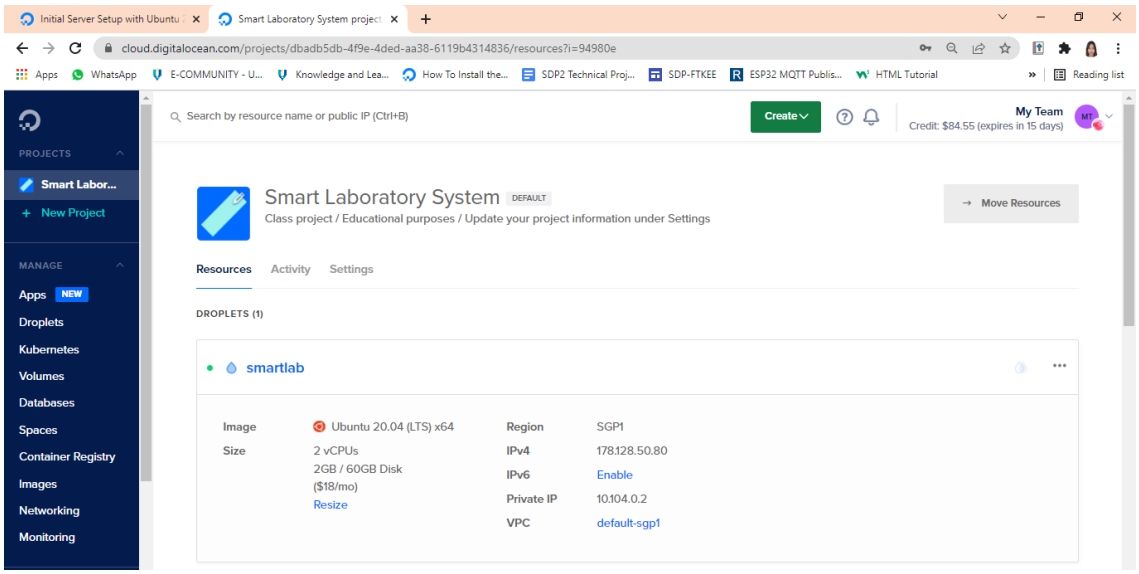


Figure 4.7: Complete Creating Droplet

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 root@178.128.50.80, enter the password as shown in Figure 4.8.

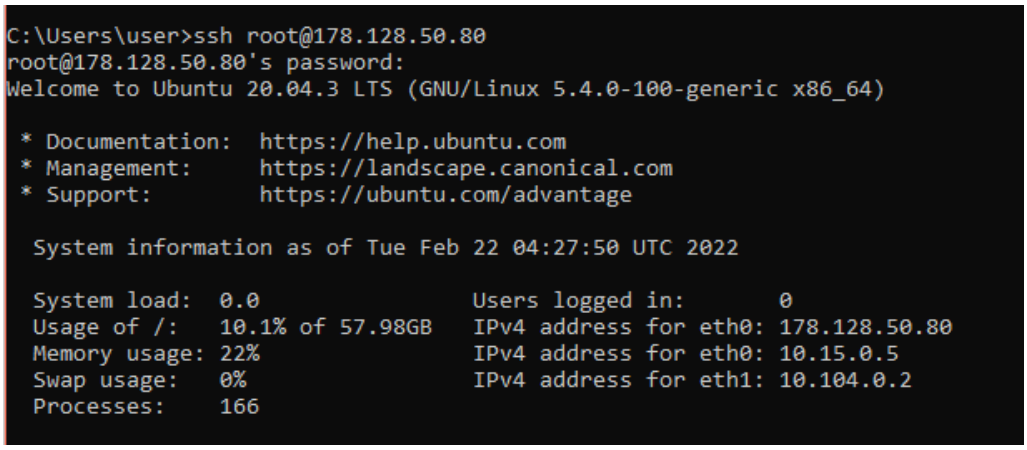


Figure 4.8: Log In as Root

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

To                         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.

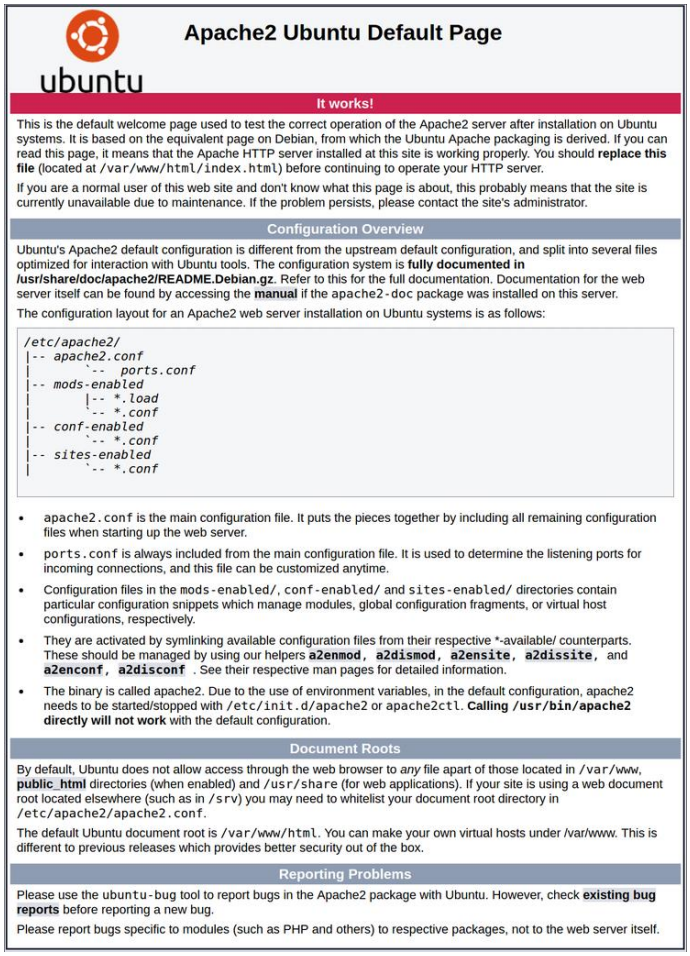


Figure 4.9: Default Ubuntu Apache Web Page

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.

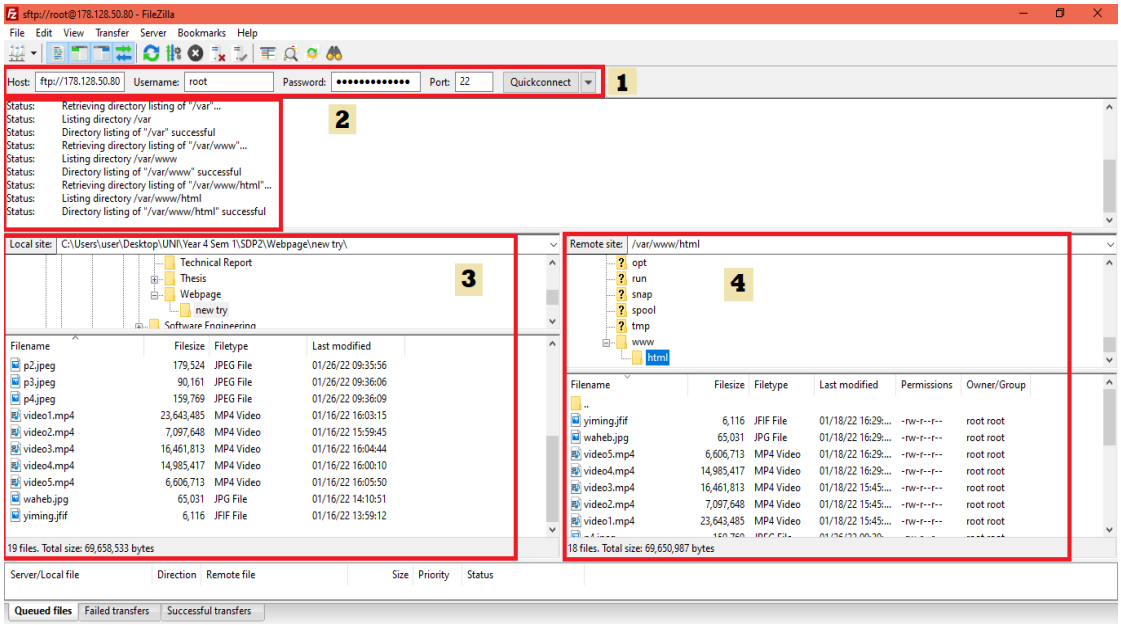


Figure 4.10: FileZilla Overview

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 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 <a href = hashtag # with the id selector> 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 <img src> 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.

index.php

```
<body>

<!-- header section starts  -->

<header>

    <div id="menu-bar" class="fas fa-bars"></div>

    <a href="#"><img src="logo.png" class="logo"></a>

    <nav class="navbar">
        <a href="#home">Home</a>
        <a href="#services">Services</a>
        <a href="#gallery">Gallery</a>
        <a href="#about">About Us</a>
    </nav>

    <div class="icons">
        <i class="fas fa-search" id="search-btn"></i>
        <i class="fas fa-user" id="login-btn"></i>
    </div>

    <form action="" class="search-bar-container">
        <input type="search" id="search-bar" placeholder="search here...">
        <label for="search-bar" class="fas fa-search"></label>
    </form>

</header>

<!-- header section ends -->
```

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.

```
<!-- login form container  -->

<div class="login-form-container">

    <i class="fas fa-times" id="form-close"></i>

    <form action="">
        <h3>login</h3>
        <input type="email" class="box" placeholder="enter your email">
        <input type="password" class="box" placeholder="enter your password">
        <input type="submit" value="login now" class="btn">
        <input type="checkbox" id="remember">
        <label for="remember">remember me</label>
        <p>forget password? <a href="#">click here</a></p>
        <p>don't have and account? <a href="#">register now</a></p>
    </form>

</div>
```

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.

```
<!-- home section starts  -->

<section class="home" id="home">

    <div class="content">
        <h1>Ump Smart Laboratory</h1><br>
        <p><h3>Everything becomes smarter and easier with the used of IOT</h3></p><br>
        <button><a href="http://192.168.2.127:1880/ui">Click to View</a></button>
    </div>

    <div class="controls">
        <span class="vid-btn active" data-src="video1.mp4"></span>
        <span class="vid-btn" data-src="video2.mp4"></span>
        <span class="vid-btn" data-src="video3.mp4"></span>
        <span class="vid-btn" data-src="video4.mp4"></span>
        <span class="vid-btn" data-src="video5.mp4"></span>
    </div>

    <div class="video-container">
        <video src="video1.mp4" id="video-slider" loop autoplay muted></video>
    </div>

</section>

<!-- home section ends -->
```

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 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.

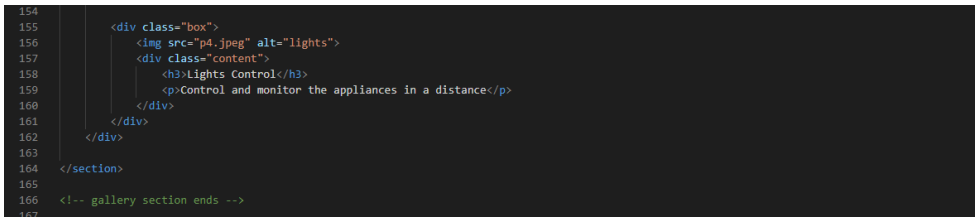
```
<!-- services section starts  -->
<section class="services" id="services">
    <h1 class="heading">services</h1>
    <div class="box-container">
        <div class="box">
            <i class="fas fa-thermometer-half"></i>
            <h3>Real Time Data</h3>
            <p>The users are able to view the real-time data temperature and humidity of the laboratory.</p>
        </div>

        <div class="box">
            <i class="fas fa-camera"></i>
            <h3>Security</h3>
            <p>The ESP32 Camera will capture the laboratory and the PIR motion sensor will detect the motion.</p>
        </div>

        <div class="box">
            <i class="fas fa-volume-up"></i>
            <h3>Safety</h3>
            <p>An SMS alert will be sent to the authorities when there is motion detected.</p>
        </div>

        <div class="box">
            <i class="fas fa-globe-asia"></i>
            <h3>Worldwide</h3>
            <p>The laboratory users are able to control and monitor the appliances even at a distance.</p>
        </div>
    </div>
</section>
<!-- services section ends -->
```

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 is used to locate the boxes of the gallery content. Four images have been put in the and displayed on the web page by using the <img src>. The gallery content is styled with h3 headings for the gallery title and normal font size for the gallery description.

```
<!-- gallery section starts  -->

<section class="gallery" id="gallery">

    <h1 class="heading">gallery</h1>
    <div class="box-container">
        <div class="box">
            <img src="p1.jpeg" alt="tem/hum">
            <div class="content">
                <h3>DHT11 Sensor</h3>
                <p>Temperature and Humidity Sensor</p>
            </div>
        </div>

        <div class="box">
            <img src="p2.jpeg" alt="camera">
            <div class="content">
                <h3>ESP32 Camera</h3>
                <p>Display the image of the laboratory situation</p>
            </div>
        </div>

        <div class="box">
            <img src="p3.jpeg" alt="motion">
            <div class="content">
                <h3>PIR Motion Sensor</h3>
                <p>To detect motion in the laboratory</p>
            </div>
        </div>
```

Figure 4.16: PHP Gallery Section 1



```
        <div class="box">
            <img src="p4.jpeg" alt="lights">
            <div class="content">
                <h3>Lights Control</h3>
                <p>Control and monitor the appliances in a distance</p>
            </div>
        </div>
    </div>

</section>

<!-- gallery section ends -->
```

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 it is style by having the ability to swipe and view the about us content. In the , there are five and to include all the project coordinator, supervisor and members simple information.

```
<!-- about us section starts  -->

<section class="about" id="about">

    <h1 class="heading">about us</h1>

    <div class="swiper-container about-slider">
        <div class="swiper-wrapper">
            <div class="swiper-slide">
                <div class="box">
                    <img src="waheb.jpg" alt="coordinator">
                    <h3>COORDINATOR</h3><br>
                    <p>Ts. Dr. Waheb Abdul Jabbar Shaif Abdullah</p>
                </div>
            </div>

            <div class="swiper-slide">
                <div class="box">
                    <img src="amran.jpg" alt="supervisor">
                    <h3>SUPERVISOR</h3><br>
                    <p>Encik Amran Bin Abdul Hadi</p>
                </div>
            </div>

            <div class="swiper-slide">
                <div class="box">
                    <img src="aqil.jfif" alt="member">
                    <h3>MEMBER</h3><br>
                    <p>Muhammad Aqil Bin Muhammad</p>
                    <p>TB18018</p>
                </div>
            </div>
```

Figure 4.18: PHP About Us Section 1

```
            <div class="swiper-slide">
                <div class="box">
                    <img src="carlos.jfif" alt="member">
                    <h3>MEMBER</h3><br>
                    <p>Carlos Enseriban</p>
                    <p>TB18036</p>
                </div>
            </div>

            <div class="swiper-slide">
                <div class="box">
                    <img src="yiming.jfif" alt="member">
                    <h3>MEMBER</h3><br>
                    <p>Lee Yi Ming</p>
                    <p>TG18040</p>
                </div>
            </div>
        </div>
    </div>
</section>

<!-- about us section ends -->
```

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.

```
<!-- footer section  -->

<section class="footer">

    <div class="box-container">
        <div class="box">
            <h3>Follow us</h3><br>
            <a href="#"><i class="fab fa-facebook"> Facebook</i></a>
            <a href="#"><i class="fab fa-instagram"> Instagram</i></a>
            <a href="#"><i class="fab fa-whatsapp"> Whatsapp</i></a>
        </div>
    </div>

</section>


<script src="https://unpkg.com/swiper/swiper-bundle.min.js"></script>

<!-- custom js file link  -->
<script src="newjavascript.js"></script>

</body>
</html>
```

Figure 4.20: PHP Footer

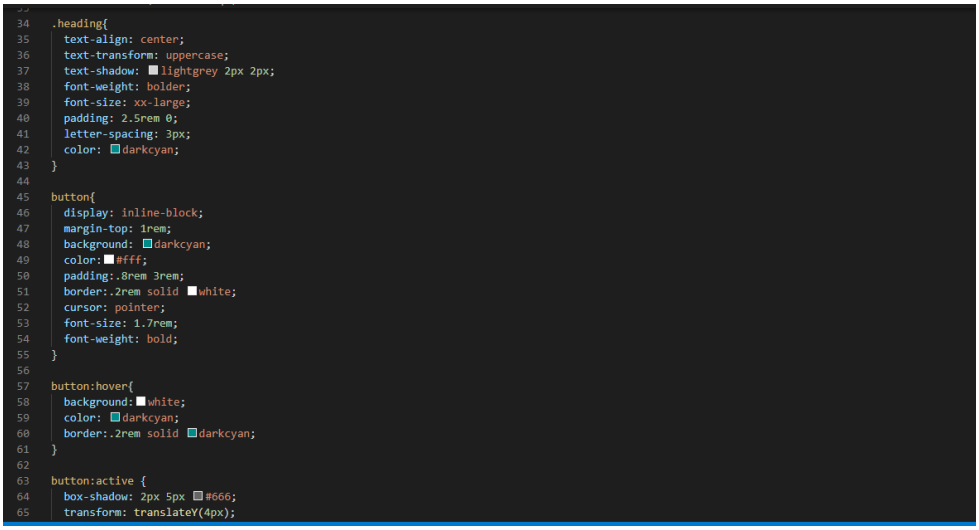
4.4.2 CSS

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.

```
:root{
  --darkcyan: darkcyan;
}

*{
  font-family: 'Verdana', sans-serif;
  margin:0; padding:0;
  box-sizing: border-box;
  outline: none;
  border:none;
  text-decoration: none;
  transition: all .2s linear;
}

*::selection{
  background:var(--darkcyan);
  color: #fff;
}

html{
  font-size: 55%;
  align-items: center;
  overflow-x: hidden;
  scroll-padding-top: 6rem;
  scroll-behavior: smooth;
}

section{
  padding:2rem 9%;
}
```

Figure 4.21: CSS Overall Section 1

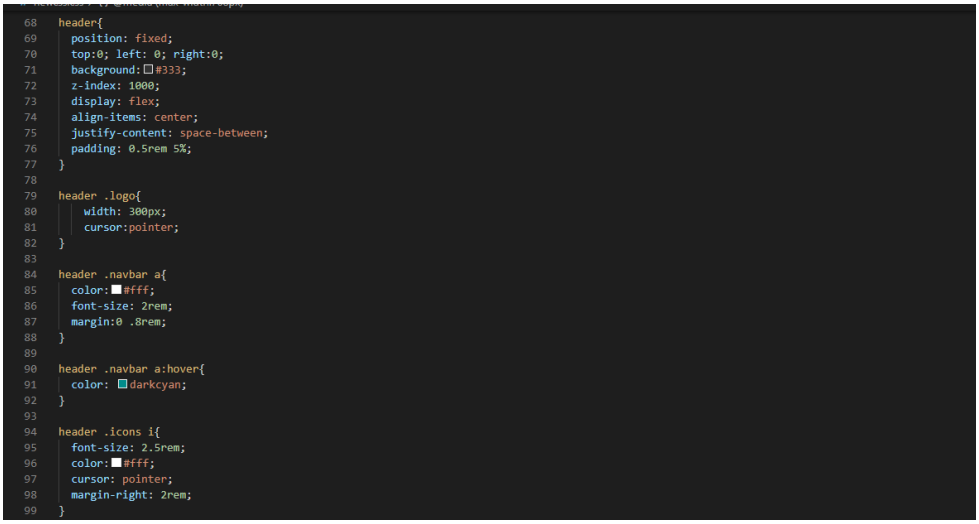


```
.heading{
  text-align: center;
  text-transform: uppercase;
  text-shadow: lightgrey 2px 2px;
  font-weight: bolder;
  font-size: xx-large;
  padding: 2.5rem 0;
  letter-spacing: 3px;
  color: darkcyan;
}

button{
  display: inline-block;
  margin-top: 1rem;
  background: darkcyan;
  color:#fff;
  padding:.8rem 3rem;
  border:.2rem solid white;
  cursor: pointer;
  font-size: 1.7rem;
  font-weight: bold;
}

button:hover{
  background:white;
  color: darkcyan;
  border:.2rem solid darkcyan;
}

button:active {
  box-shadow: 2px 5px #666;
  transform: translateY(4px);
```

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.



```
header{
  position: fixed;
  top:0; left: 0; right:0;
  background:#333;
  z-index: 1000;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0.5rem 5%;
}

header .logo{
    width: 300px;
    cursor:pointer;
}

header .navbar a{
  color:#fff;
  font-size: 2rem;
  margin:0 .8rem;
}

header .navbar a:hover{
  color: darkcyan;
}

header .icons i{
  font-size: 2.5rem;
  color:#fff;
  cursor: pointer;
  margin-right: 2rem;
}
```

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.

```
header .icons i:hover{
  color:var(--darkcyan);
}

header .search-bar-container{
  position: absolute;
  top:100%; left: 0; right:0;
  padding:1.5rem 2rem;
  background:#333;
  border-top: .1rem solid rgba(255,255,255,.2);
  display: flex;
  align-items: center;
  z-index: 1001;
  clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
}

header .search-bar-container.active{
  clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);
}

header .search-bar-container #search-bar{
  width:100%;
  padding:1rem;
  text-transform: none;
  color:#333;
  font-size: 1.7rem;
  border-radius: .5rem;
}
```

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.

```
header .search-bar-container label{
  color:#fff;
  cursor: pointer;
  font-size: 3rem;
  margin-left: 1.5rem;
}

header .search-bar-container label:hover{
  color:var(--darkcyan);
}

.login-form-container{
  position: fixed;
  top:-120%; left: 0;
  z-index: 10000;
  min-height: 100vh;
  width:100%;
  background:rgba(0,0,0,.7);
  display: flex;
  align-items: center;
  justify-content: center;
}

.login-form-container.active{
  top:0;
}
```

Figure 4.25: CSS Header Login Form Section 1

```
.login-form-container form{
  margin:2rem;
  padding:1.5rem 2rem;
  border-radius: .5rem;
  background:#fff;
  width:50rem;
}

.login-form-container form h3{
  font-size: 3rem;
  color:#444;
  text-transform: uppercase;
  text-align: center;
  padding:1rem 0;
}

.login-form-container form .box{
  width:100%;
  padding:1rem;
  font-size: 1.7rem;
  color:#333;
  margin:.6rem 0;
  border:.1rem solid rgba(0,0,0,.3);
  text-transform: none;
}

.login-form-container form .box:focus{
  border-color: var(--darkcyan);;
}
```

Figure 4.26: CSS Header Login Form Section 2

```css
.login-form-container form #remember{
  margin:2rem 0;
}

.login-form-container form label{
  font-size: 1.5rem;
}

.login-form-container form .btn{
  display: block;
  width:100%;
}

.login-form-container form p{
  padding:.5rem 0;
  font-size: 1.5rem;
  color:#666;
}

.login-form-container form p a{
  color:var(--darkcyan);
}

.login-form-container form p a:hover{
  color:#333;
  text-decoration: underline;
}
```

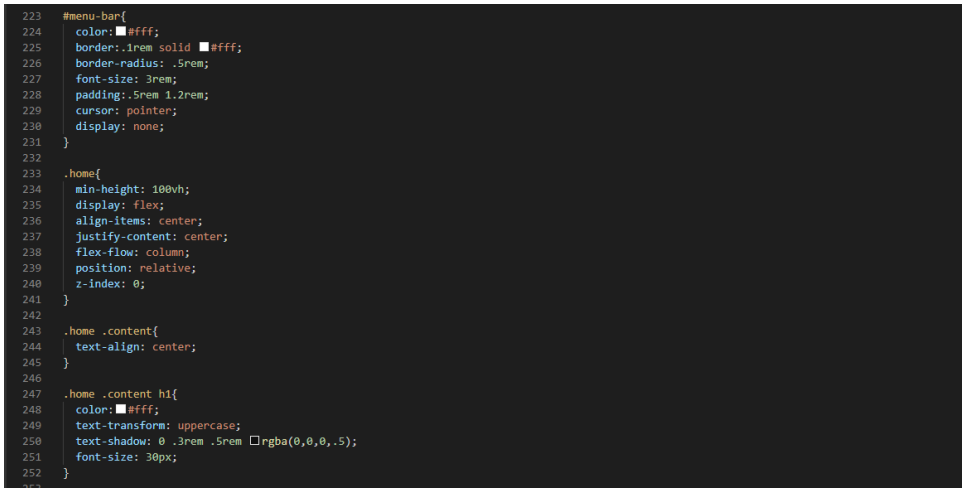
Figure 4.27: CSS Header Login Form Section 3

```css
.login-form-container #form-close{
  position: absolute;
  top:2rem; right:3rem;
  font-size: 5rem;
  color:#fff;
  cursor: pointer;
}
```

Figure 4.28: CSS Header Login Form Section 4

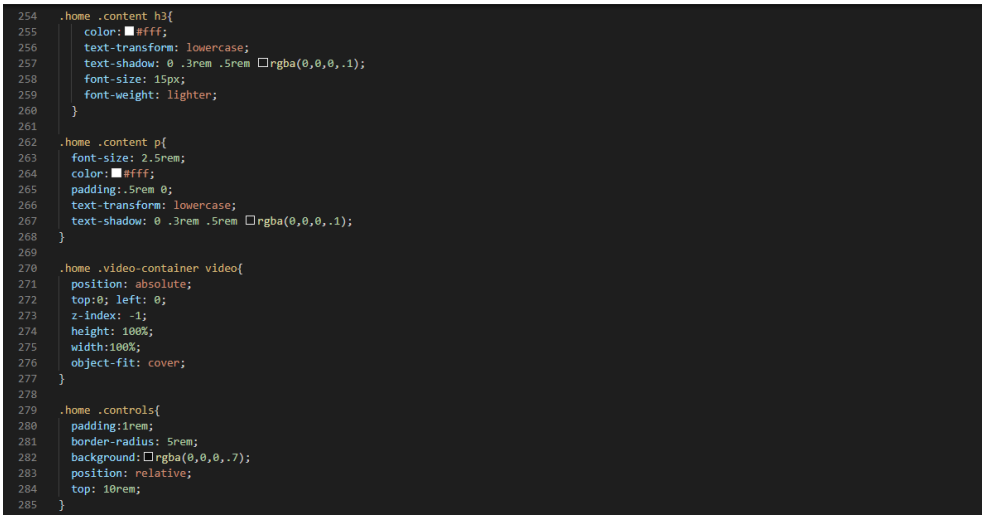
Section Home: The Home section of scripting is shown in Figure 4.29, Figure 4.30, and Figure 4.31. All of the contents of the home are placed at the center with a height of 100vh, which will be displayed on the whole screen. The tile of "UMP Smart Laboratory" is styled by using the h1 heading, uppercase 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.



```
#menu-bar{
  color:#fff;
  border:.1rem solid #fff;
  border-radius: .5rem;
  font-size: 3rem;
  padding:.5rem 1.2rem;
  cursor: pointer;
  display: none;
}

.home{
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-flow: column;
  position: relative;
  z-index: 0;
}

.home .content{
  text-align: center;
}

.home .content h1{
  color:#fff;
  text-transform: uppercase;
  text-shadow: 0 .3rem .5rem rgba(0,0,0,.5);
  font-size: 30px;
}
```

Figure 4.29: CSS Home Section 1



```
.home .content h3{
    color:#fff;
    text-transform: lowercase;
    text-shadow: 0 .3rem .5rem rgba(0,0,0,.1);
    font-size: 15px;
    font-weight: lighter;
}

.home .content p{
  font-size: 2.5rem;
  color:#fff;
  padding:.5rem 0;
  text-transform: lowercase;
  text-shadow: 0 .3rem .5rem rgba(0,0,0,.1);
}

.home .video-container video{
  position: absolute;
  top:0; left: 0;
  z-index: -1;
  height: 100%;
  width:100%;
  object-fit: cover;
}

.home .controls{
  padding:1rem;
  border-radius: 5rem;
  background:rgba(0,0,0,.7);
  position: relative;
  top: 10rem;
}
```

Figure 4.30: CSS Home Section 2

```
.home .controls .vid-btn{
  height:2rem;
  width:2rem;
  display: inline-block;
  border-radius: 50%;
  background:#fff;
  cursor: pointer;
  margin:0 .5rem;
}

.home .controls .vid-btn.active{
  background: darkcyan;
}
```

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.

```
.services .box-container{
  display: flex;
  flex-wrap: wrap;
  gap: 1.8rem;
}

.services .box-container .box{
  flex: 1 1 30rem;
  border-radius: 1.5rem;
  padding: 3rem 2rem;
  text-align: center;
  background-color: darkcyan;
}

.services .box-container .box i{
  padding:1rem;
  font-size: 5rem;
  color: wheat;
  text-shadow: 4px 2px darkgray;
}

.services .box-container .box h3{
  font-size: 2.5rem;
  color: wheat;
}
```

Figure 4.32: CSS Services Section 1

```
.services .box-container .box p{
  font-size: 1.5rem;
  color:white;
  padding: 2rem 0;
  text-align: center;
}

.services .box-container .box:hover{
  box-shadow: 0 1rem 2rem rgba(0,0,0,.4);
  transform: scale(1.1);
}
```

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.

```css
.gallery .box-container{
  display: flex;
  flex-wrap: wrap;
  gap: 1.5rem;
}

.gallery .box-container .box{
  overflow: hidden;
  box-shadow: 0 1rem 2rem rgba(0,0,0,.5);
  border: 1rem solid #fff;
  border-radius: .5rem;
  flex:1 1 30rem;
  height: 30rem;
  position: relative;
}

.gallery .box-container .box img{
  height: 100%;
  width: 100%;
  object-fit: cover;
}

.gallery .box-container .box .content{
  position: absolute;
  top:-100%; left:0;
  height: 100%;
  width:100%;
  text-align: center;
  background:rgba(0,0,0,.45);
  padding: 2rem;
  padding-top: 10rem;
}
```

Figure 4.34: CSS Gallery Section 1

```css
.gallery .box-container .box:hover .content{
  top:0;
}

.gallery .box-container .box .content h3{
  font-size: 2.5rem;
  color: wheat;
}

.gallery .box-container .box .content p{
  font-size: 1.5rem;
  color:#eee;
  padding: 1rem 0;
  text-transform: lowercase;
}
```

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

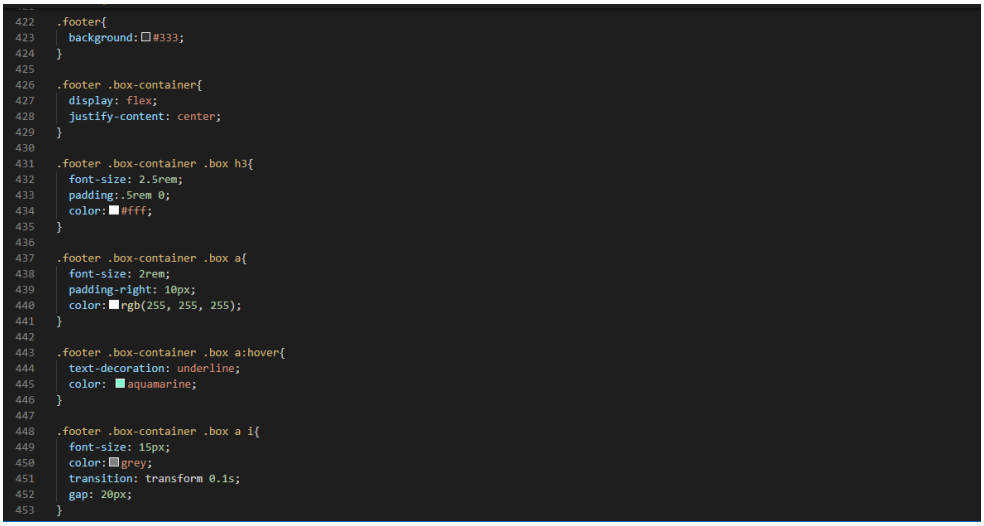
font size, which is 1.5rem. The user can scroll to the left or right to view the profile content.

```
.about .about-slider{
  padding-bottom: 2rem;
}

.about .box{
  padding: 1rem;
  text-align: center;
  box-shadow: 0 1rem 2rem rgba(0,0,0,.5);
  border-radius: 5px;
  background-color: lightgrey;
  height: 40vh;
}

.about .box img{
  height:15rem;
  width:15rem;
  border-radius: 50%;
  object-fit: cover;
  margin-top: 2rem;
  margin-bottom: 2rem;
}

.about .box h3{
  color: darkslategray;
  text-transform: uppercase;
  font-weight: bold;
  font-size: 2.0rem;
}

.about .box p{
  color: black;
  font-size: 1.5rem;
```

Figure 4.36: CSS About Us Section

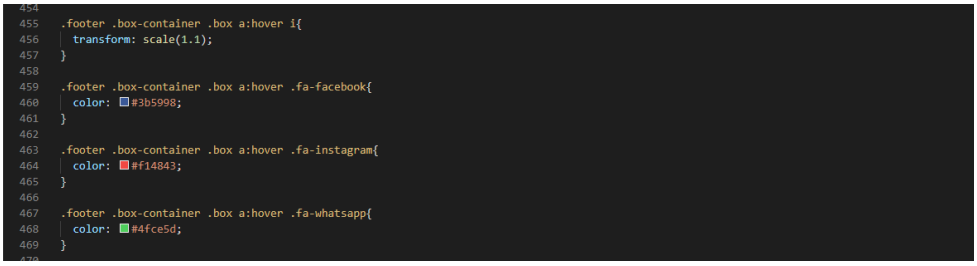
Section Footer: This section's scripting is as shown in Figure 4.37 and Figure 4.38. Footer is used to directing the users to follow us on social media such as Facebook, Instagram, and WhatsApp. The footer background color is slightly lighter grey than the navbar background color. The content in this 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.



```
.footer{
  background:#333;
}

.footer .box-container{
  display: flex;
  justify-content: center;
}

.footer .box-container .box h3{
  font-size: 2.5rem;
  padding:.5rem 0;
  color:#fff;
}

.footer .box-container .box a{
  font-size: 2rem;
  padding-right: 10px;
  color:rgb(255, 255, 255);
}

.footer .box-container .box a:hover{
  text-decoration: underline;
  color: aquamarine;
}

.footer .box-container .box a i{
  font-size: 15px;
  color:grey;
  transition: transform 0.1s;
  gap: 20px;
}
```

Figure 4.37: CSS Footer Section 1



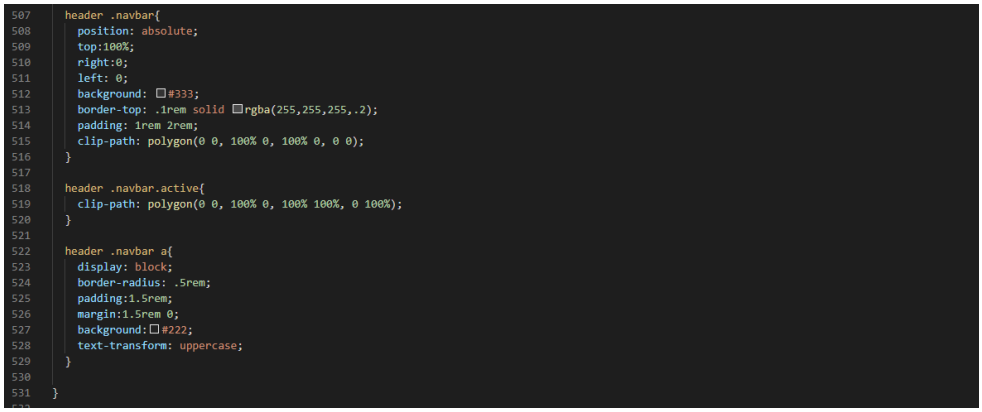
```
.footer .box-container .box a:hover i{
  transform: scale(1.1);
}

.footer .box-container .box a:hover .fa-facebook{
  color: #3b5998;
}

.footer .box-container .box a:hover .fa-instagram{
  color: #f14843;
}

.footer .box-container .box a:hover .fa-whatsapp{
  color: #4fce5d;
}
```

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.

```
/* media queries  */

@media (max-width:1200px){

  html{
    font-size: 55%;
  }

}

@media (max-width:991px){

  header{
    padding:2rem;
  }

  section{
    padding:2rem;
  }

}

@media (max-width:768px){

  #menu-bar{
    display: initial;
  }
```

Figure 4.39: CSS Media Queries Section 1



```
header .navbar{
  position: absolute;
  top:100%;
  right:0;
  left: 0;
  background: #333;
  border-top: .1rem solid rgba(255,255,255,.2);
  padding: 1rem 2rem;
  clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
}

header .navbar.active{
  clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);
}

header .navbar a{
  display: block;
  border-radius: .5rem;
  padding:1.5rem;
  margin:1.5rem 0;
  background:#222;
  text-transform: uppercase;
}

}
```

Figure 4.40: CSS Media Queries Section 3

```
@media (max-width:450px){

  html{
    font-size: 50%;
  }

  .heading span{
    font-size: 2.5rem;
  }

  .contact .row form .inputBox input{
    width:100%;
  }

}
```

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.

```
let searchBtn = document.querySelector('#search-btn');
let searchBar = document.querySelector('.search-bar-container');
let formBtn = document.querySelector('#login-btn');
let loginForm = document.querySelector('.login-form-container');
let formClose = document.querySelector('#form-close');
let menu = document.querySelector('#menu-bar');
let navbar = document.querySelector('.navbar');
let videoBtn = document.querySelectorAll('.vid-btn');

window.onscroll = () =>{
    searchBtn.classList.remove('fa-times');
    searchBar.classList.remove('active');
    menu.classList.remove('fa-times');
    navbar.classList.remove('active');
    loginForm.classList.remove('active');
}

menu.addEventListener('click', () =>{
    menu.classList.toggle('fa-times');
    navbar.classList.toggle('active');
});

searchBtn.addEventListener('click', () =>{
    searchBtn.classList.toggle('fa-times');
    searchBar.classList.toggle('active');
});

formBtn.addEventListener('click', () =>{
    loginForm.classList.add('active');
});
```

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.

```
formClose.addEventListener('click', () =>{
    loginForm.classList.remove('active');
});

videoBtn.forEach(btn =>{
    btn.addEventListener('click', ()=>{
        document.querySelector('.controls .active').classList.remove('active');
        btn.classList.add('active');
        let src = btn.getAttribute('data-src');
        document.querySelector('#video-slider').src = src;
    });
});

var swiper = new Swiper(".about-slider", {
    spaceBetween: 50,
    loop:true,
    autoplay: {
        delay: 2800,
        disableOnInteraction: false,
    },
    breakpoints: {
        640: {
          slidesPerView: 1,
        },
        768: {
          slidesPerView: 2,
        },
        1024: {
          slidesPerView: 3,
        },
    },
});
```

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.

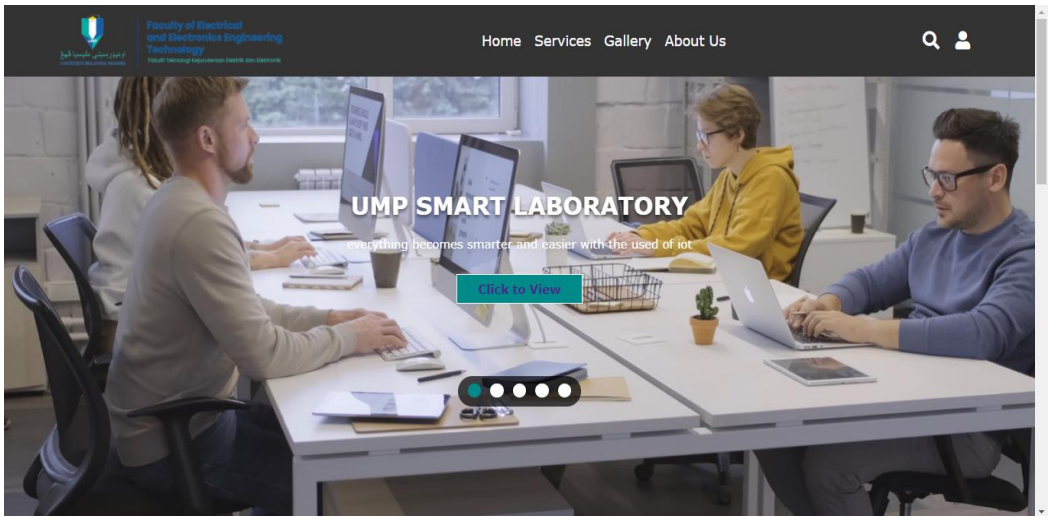


Figure 4.44: Smart Laboratory System Home Web Page

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.

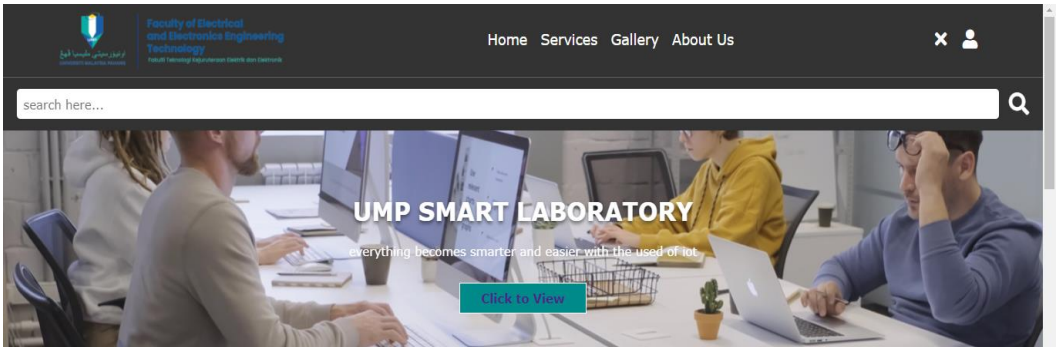


Figure 4.45: Search Bar Outcome

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.

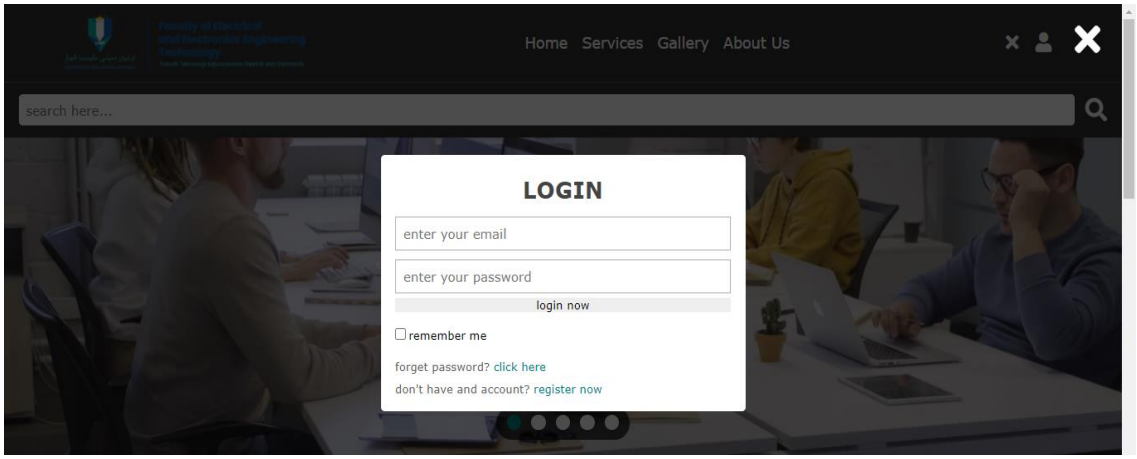


Figure 4.46: Login Form Outcome

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

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.

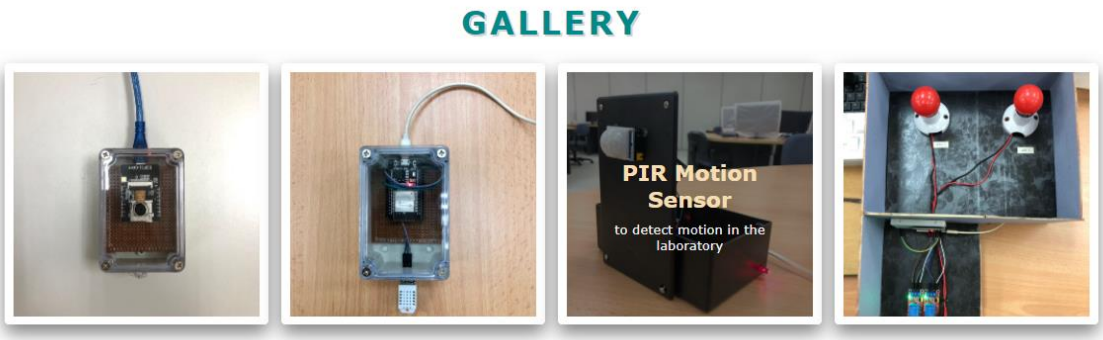


Figure 4.48: Web Page Part Gallery

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.

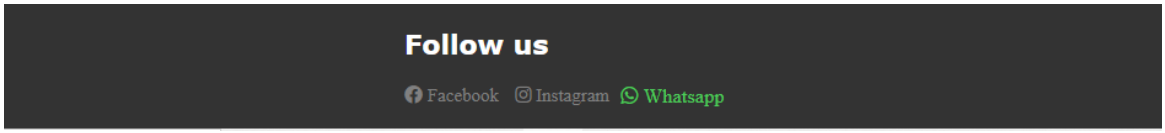


Figure 4.50: Web Page Footer

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.

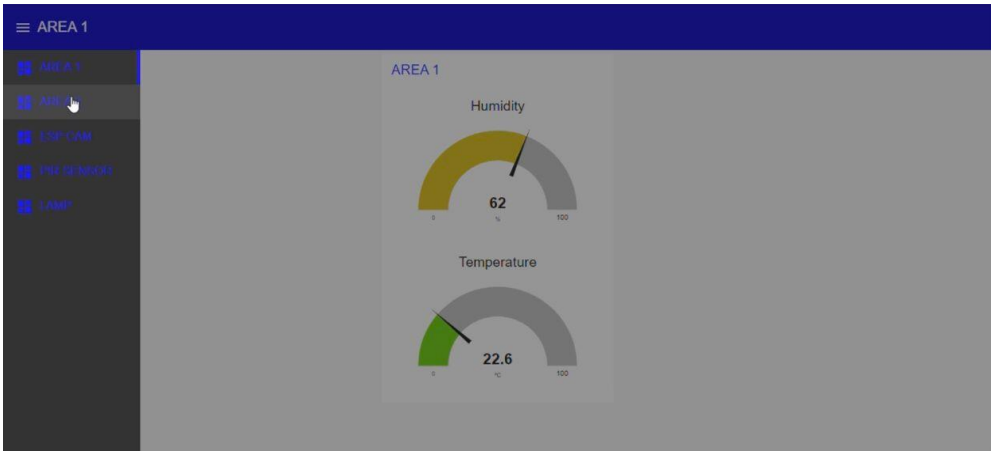


Figure 4.51: Side navigation bar

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.

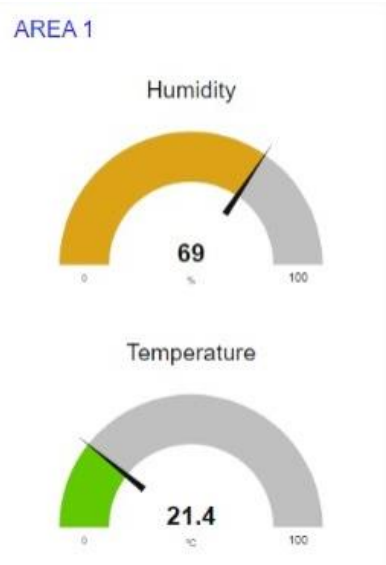


Figure 4.52: Temperature and Humidity in Area 1 of Laboratory

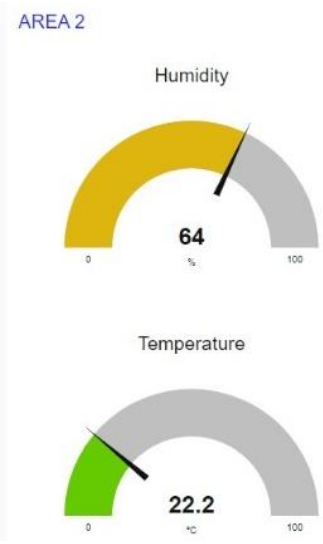


Figure 4.53: Temperature and Humidity in Area 2 of Laboratory

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 4.55: No Motion Deteced

Figure 4.56: Motion Detected

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

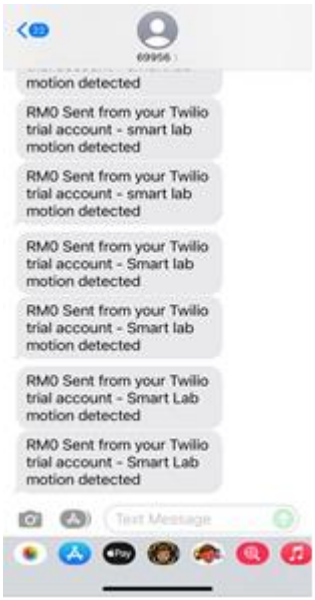


Figure 4.57: Message Alert from Twilio

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.

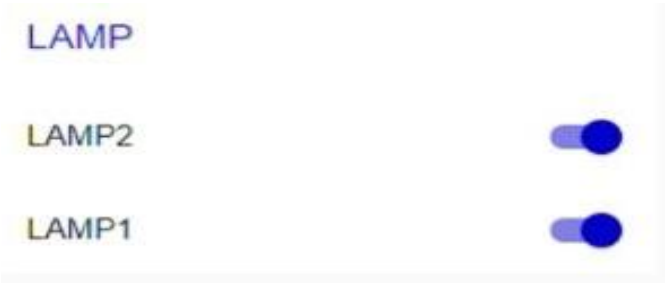


Figure 4.58: Lamp Switches

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

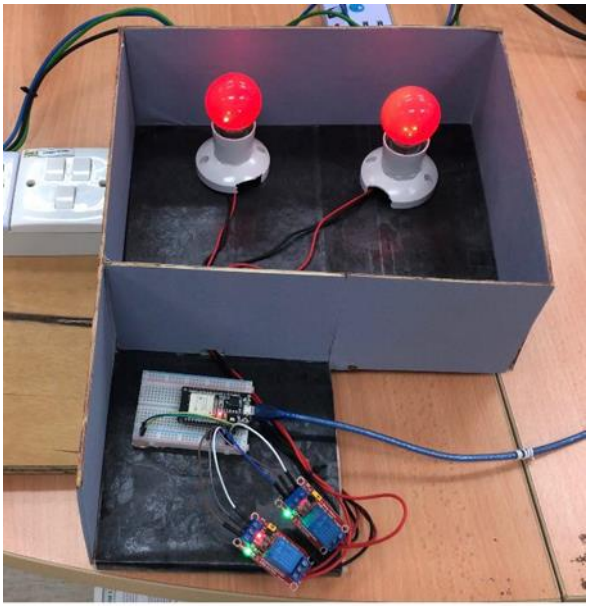


Figure 4.59: Lamp Turned On

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.

5

CONCLUSION

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.

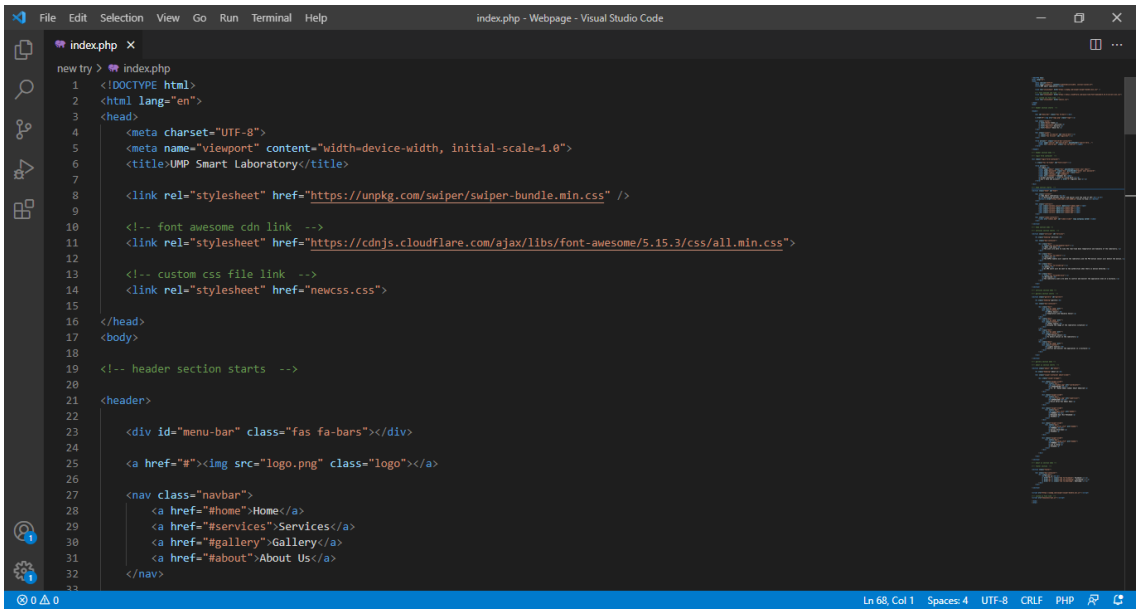
Sriskanthan, Nadarajah, Forest Tan, and Advait Karande. 2002. “Bluetooth Based Home Automation System.” *Microprocessors and Microsystems* 26(6):281–89. doi: 10.1016/S0141-9331(02)00039-X.

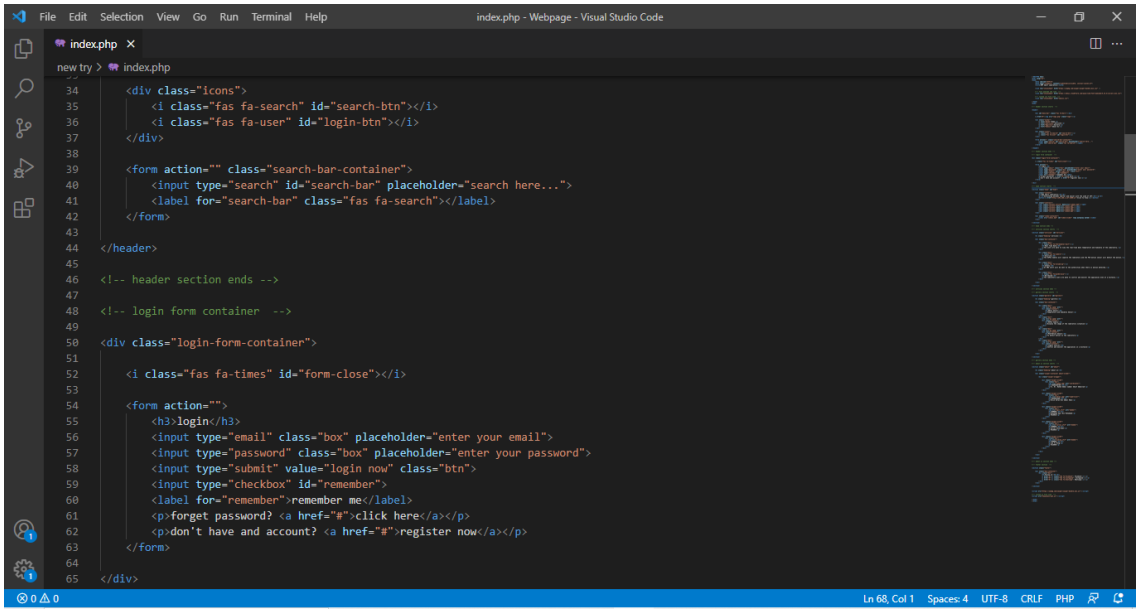
APPENDICES

Appendix A: Gantt Chart for Senior Project

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

Appendix B: PHP Coding

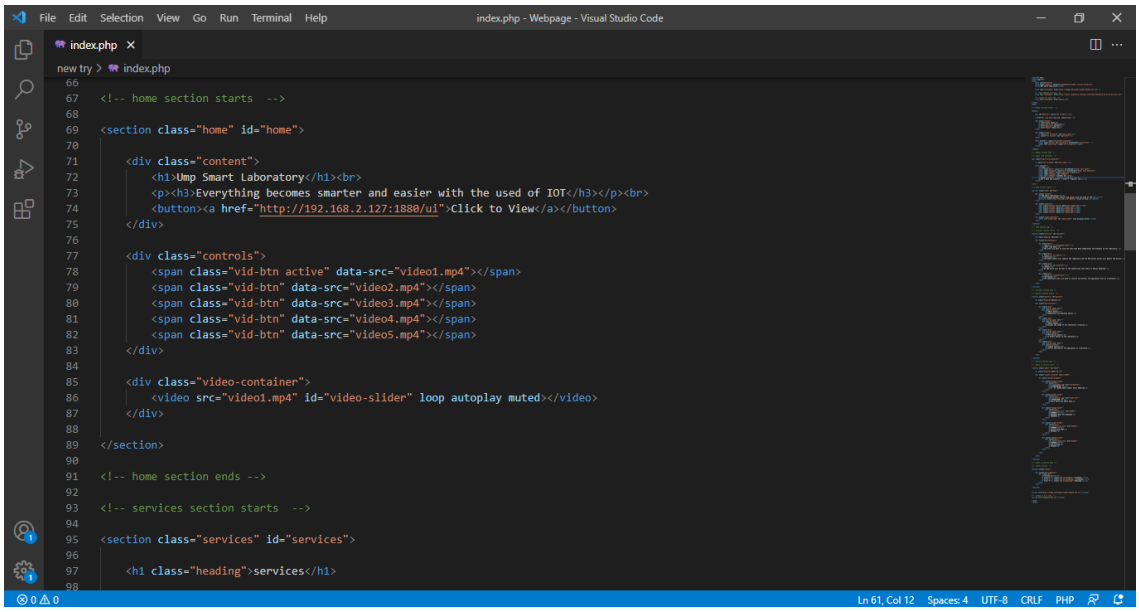



File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php

<form action="" class="search-bar-container">
<input type="search" id="search-bar" placeholder="search here...">
<label for="search-bar" class="fas fa-search"></label>
</form>
</header>
<!-- header section ends -->
<!-- login form container -->

<form action="">
<h3>login</h3>
<input type="email" class="box" placeholder="enter your email">
<input type="password" class="box" placeholder="enter your password">
<input type="submit" value="login now" class="btn">
<input type="checkbox" id="remember">
<label for="remember">remember me</label>
forget password? click here
don't have and account? register now
</form>

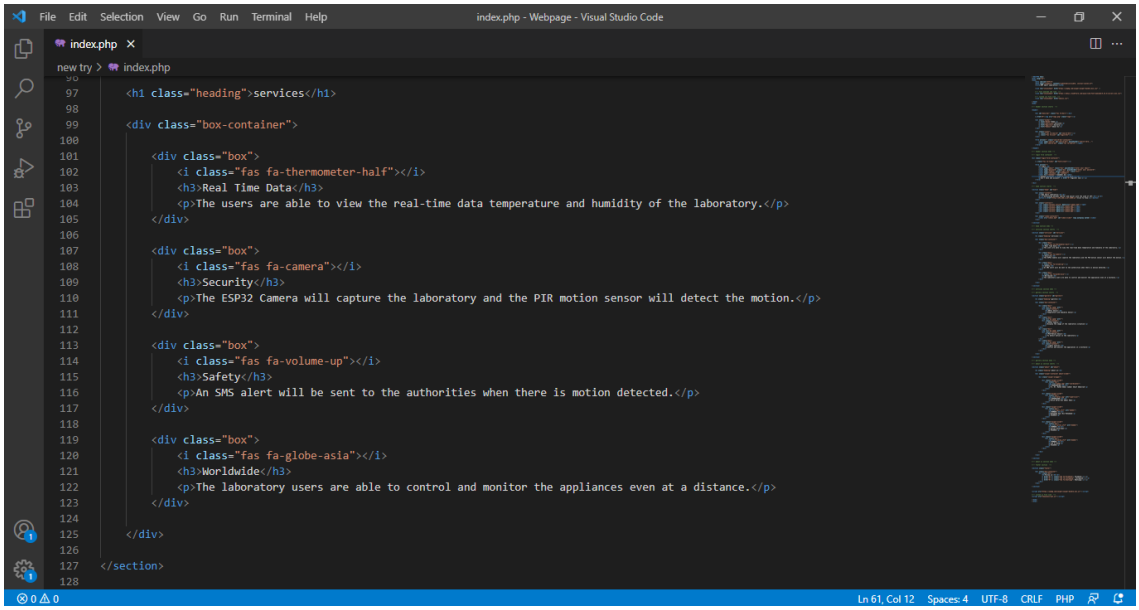
Ln 68, Col 1 Spaces: 4 UTF-8 CRLF PHP

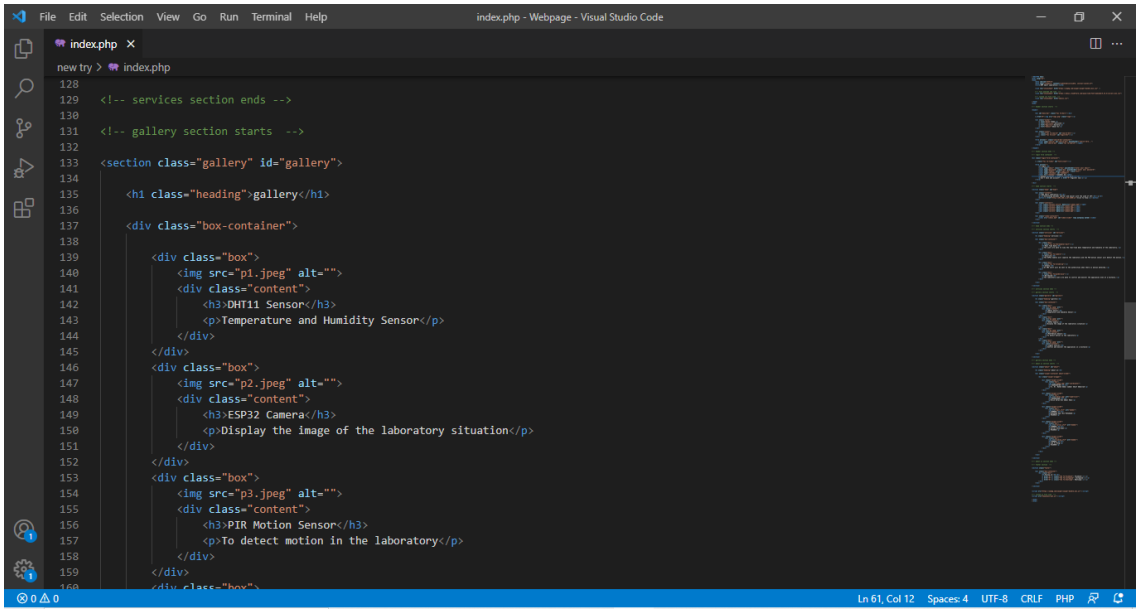

File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php
<!-- home section starts -->
<section class="home" id="home">

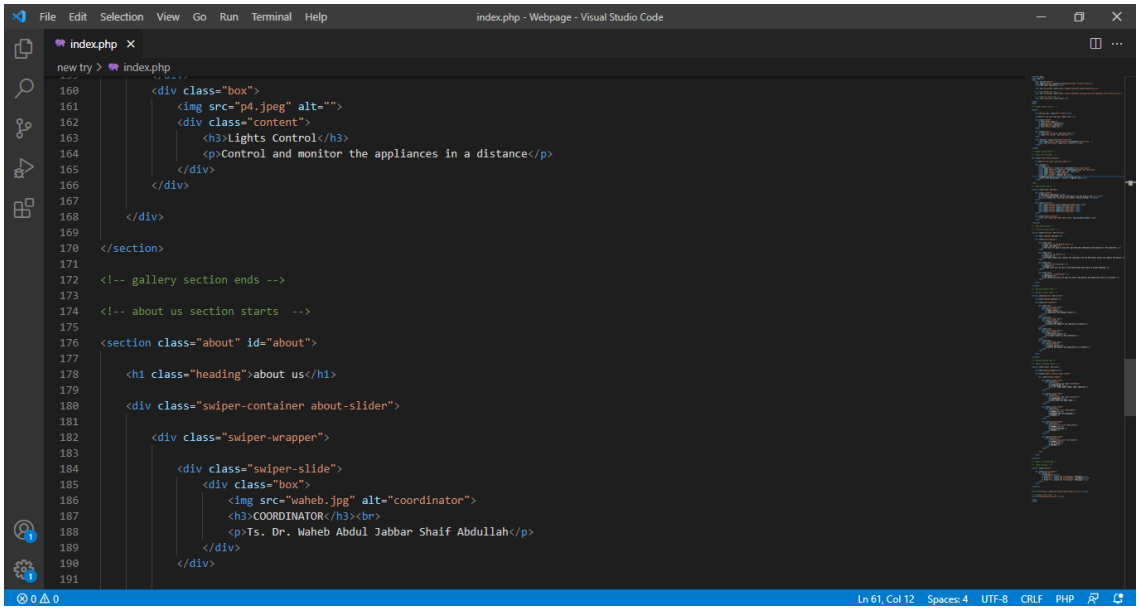
<h1>Ump Smart Laboratory</h1>
<h3>Everything becomes smarter and easier with the used of IOT</h3>
<button>Click to View</button>

<video src="video1.mp4" id="video-slider" loop autoplay muted></video>

</section>
<!-- home section ends -->
<!-- services section starts -->
<section class="services" id="services">
<h1 class="heading">services</h1>
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP


File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php
97 <h1 class="heading">services</h1>
99
101
102
103 <h3>Real Time Data</h3>
104 The users are able to view the real-time data temperature and humidity of the laboratory.
105
107
108
109 <h3>Security</h3>
110 The ESP32 Camera will capture the laboratory and the PIR motion sensor will detect the motion.
111
113
114
115 <h3>Safety</h3>
116 An SMS alert will be sent to the authorities when there is motion detected.
117
119
120
121 <h3>Worldwide</h3>
122 The laboratory users are able to control and monitor the appliances even at a distance.
123
125
127 </section>
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP


File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php
129 <!-- services section ends -->
131 <!-- gallery section starts -->
133 <section class="gallery" id="gallery">
135 <h1 class="heading">gallery</h1>
137
139
140 <img src="p1.jpeg" alt="">
141
142 <h3>DHT11 Sensor</h3>
143 Temperature and Humidity Sensor
144
145
146
147 <img src="p2.jpeg" alt="">
148
149 <h3>ESP32 Camera</h3>
150 Display the image of the laboratory situation
151
152
153
154 <img src="p3.jpeg" alt="">
155
156 <h3>PIR Motion Sensor</h3>
157 To detect motion in the laboratory
158
159
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP


File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php

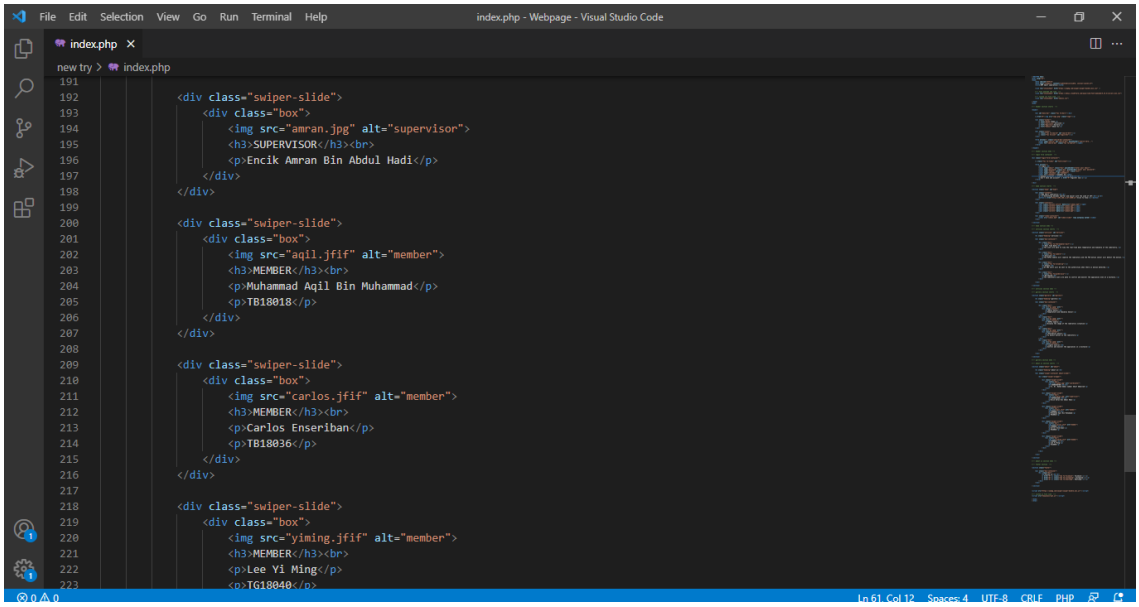
<img src="p4.jpeg" alt="">

<h3>Lights Control</h3>
Control and monitor the appliances in a distance

</section>
<!-- gallery section ends -->
<!-- about us section starts -->
<section class="about" id="about">
<h1 class="heading">about us</h1>

<img src="waheb.jpg" alt="coordinator">
<h3>COORDINATOR</h3>
Ts. Dr. Waheb Abdul Jabbar Shaif Abdullah

Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP

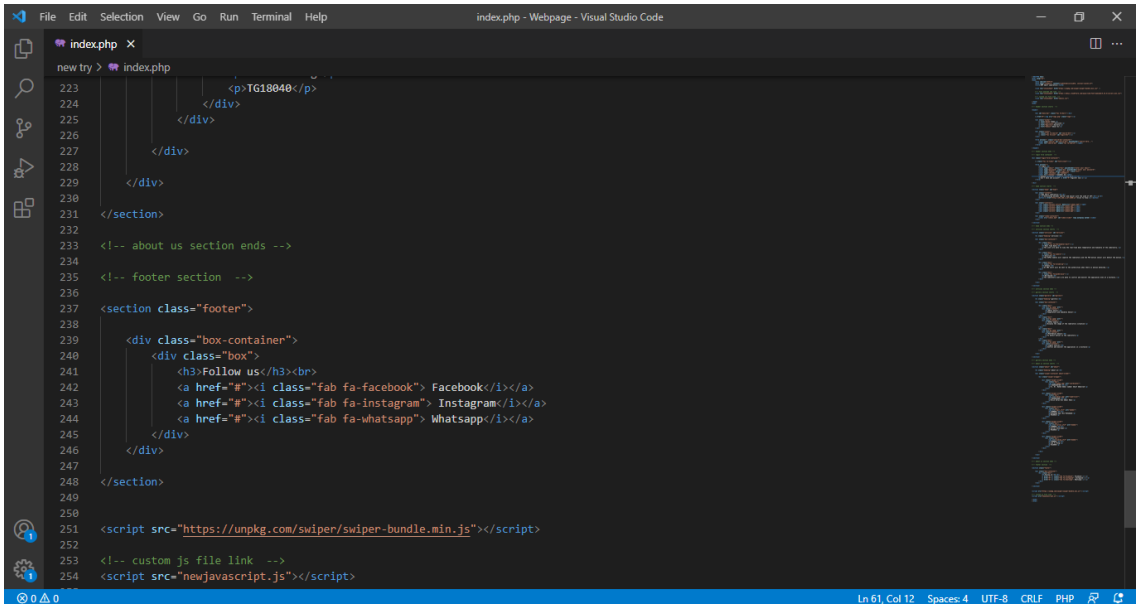

File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php

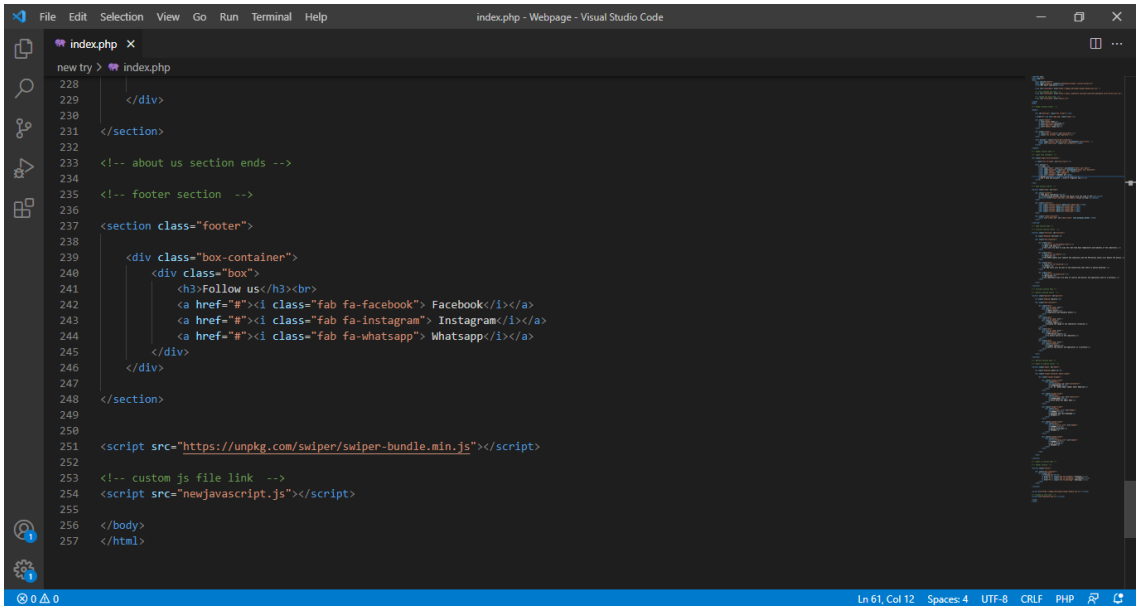
<img src="amran.jpg" alt="supervisor">
<h3>SUPERVISOR</h3>
Encik Amran Bin Abdul Hadi

<img src="aqil.jfif" alt="member">
<h3>MEMBER</h3>
Muhammad Aqil Bin Muhammad
TB18018

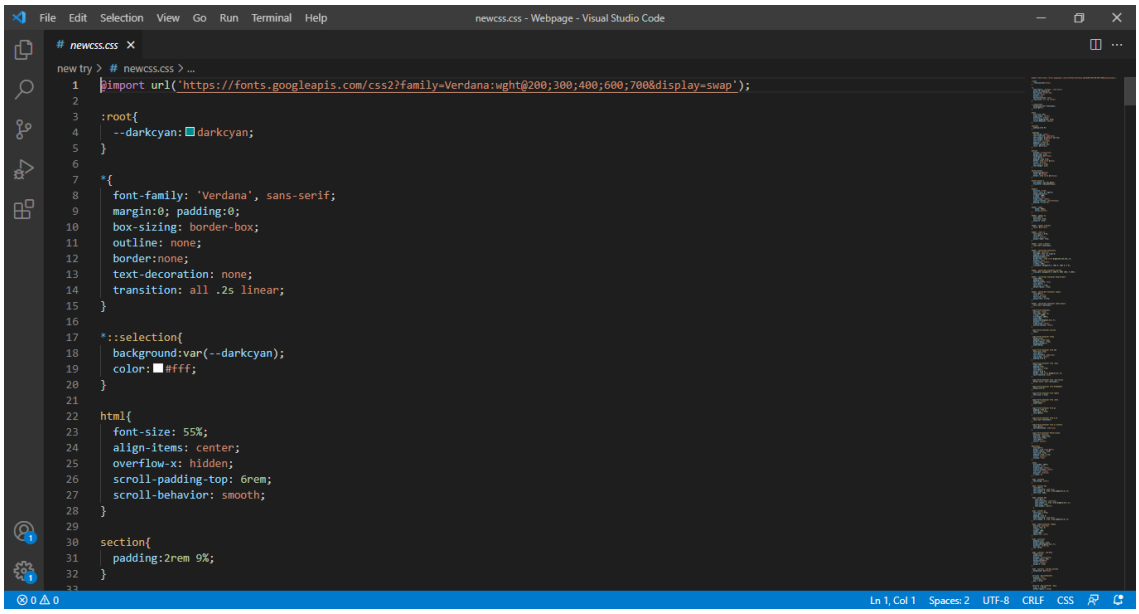
<img src="carlos.jfif" alt="member">
<h3>MEMBER</h3>
Carlos Enseriban
TB18036

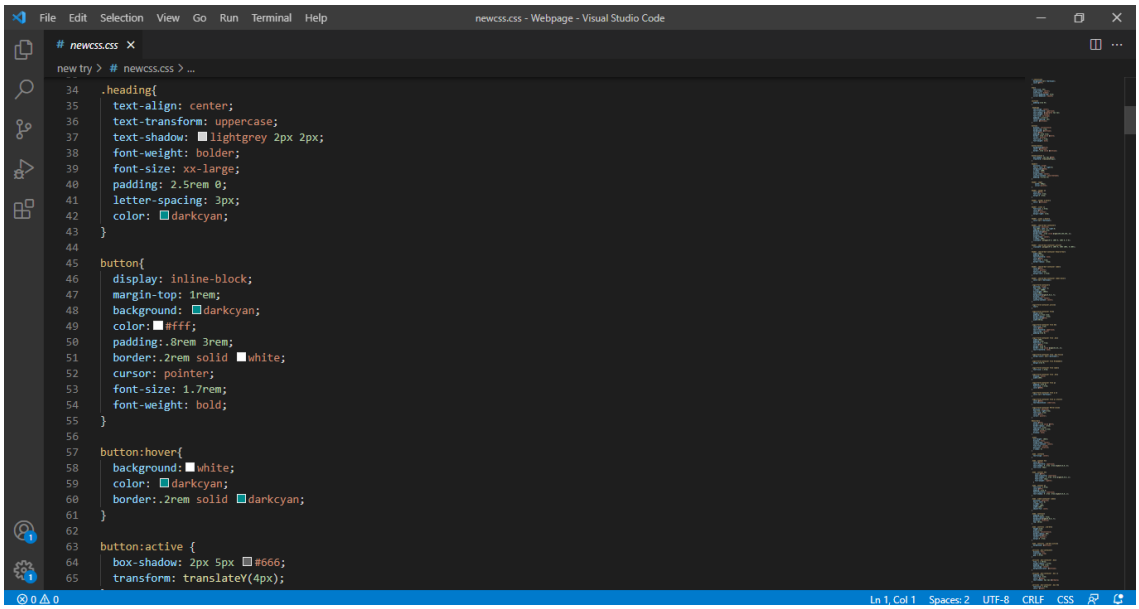
<img src="yiming.jfif" alt="member">
<h3>MEMBER</h3>
Lee Yi Ming
TG18040
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP

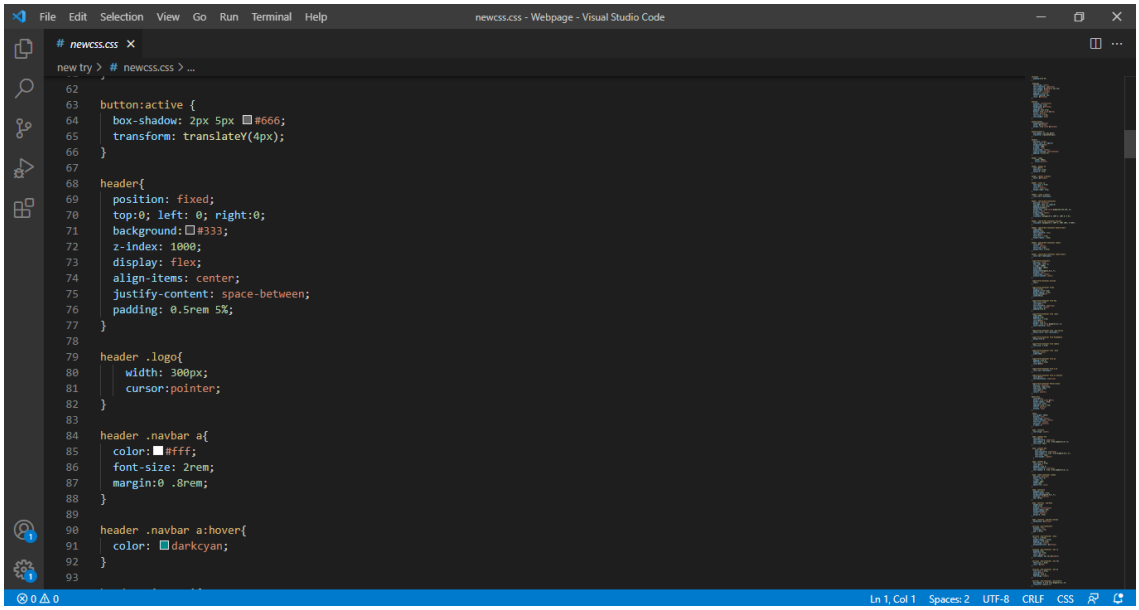

File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php
223 TG18040
224
225
226
227
228
229
230
231 </section>
232
233 <!-- about us section ends -->
234
235 <!-- footer section -->
236
237 <section class="footer">
238
239
240
241 <h3>Follow us</h3>
242 Facebook
243 Instagram
244 Whatsapp
245
246
247
248 </section>
249
250
251 <script src="https://unpkg.com/swiper/swiper-bundle.min.js"></script>
252
253 <!-- custom js file link -->
254 <script src="newjavascript.js"></script>
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP

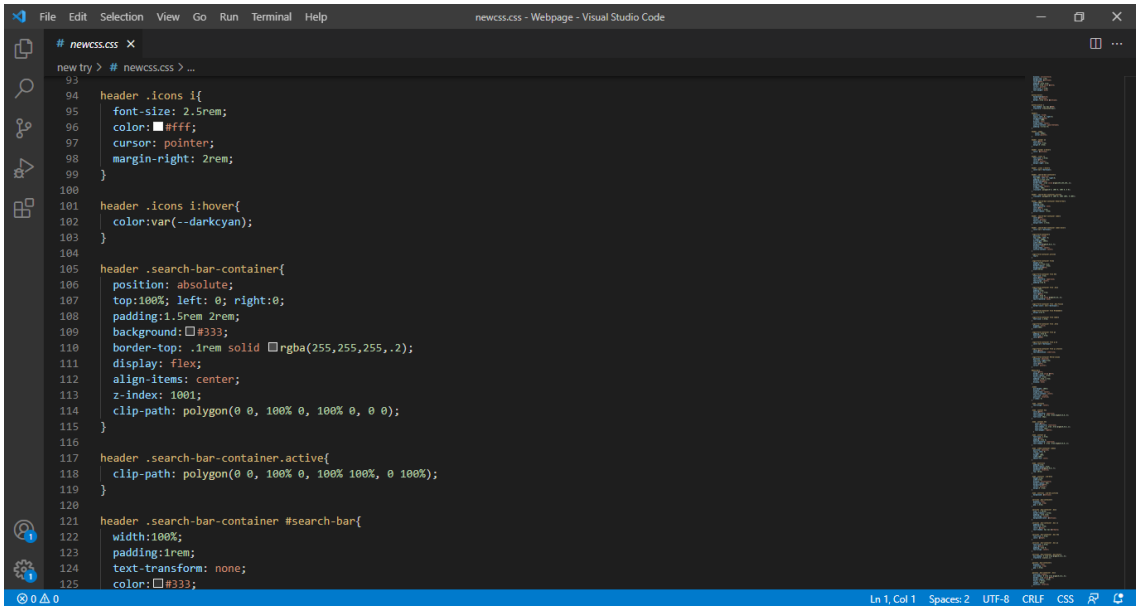

File Edit Selection View Go Run Terminal Help
index.php - Webpage - Visual Studio Code
index.php
new try > index.php
228
229
230
231 </section>
232
233 <!-- about us section ends -->
234
235 <!-- footer section -->
236
237 <section class="footer">
238
239
240
241 <h3>Follow us</h3>
242 Facebook
243 Instagram
244 Whatsapp
245
246
247
248 </section>
249
250
251 <script src="https://unpkg.com/swiper/swiper-bundle.min.js"></script>
252
253 <!-- custom js file link -->
254 <script src="newjavascript.js"></script>
255
256 </body>
257 </html>
Ln 61, Col 12 Spaces: 4 UTF-8 CRLF PHP

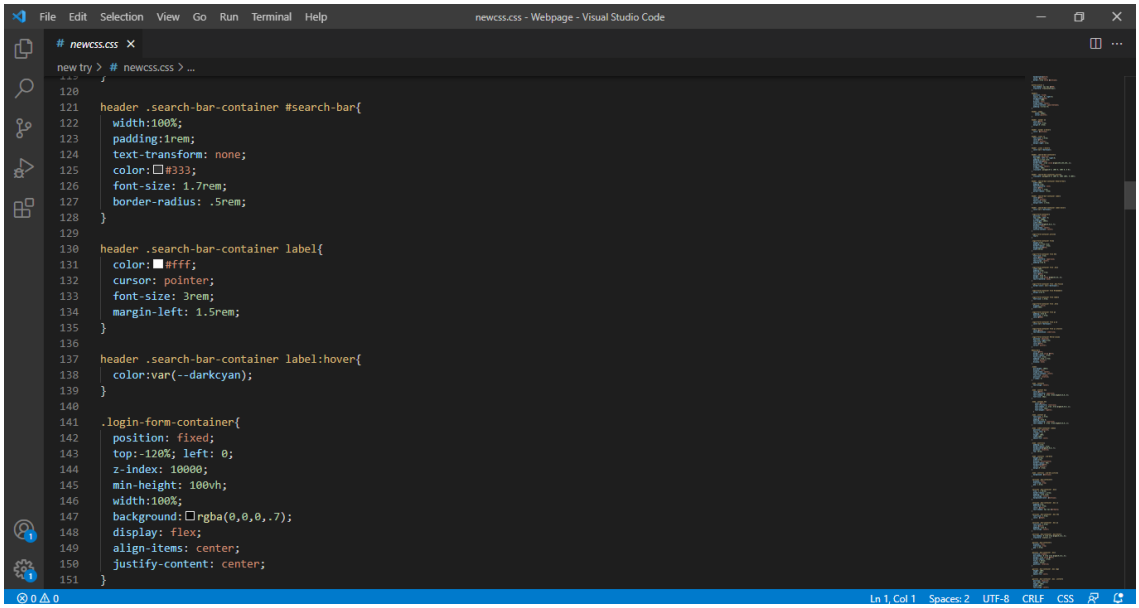
Appendix C : CSS Coding

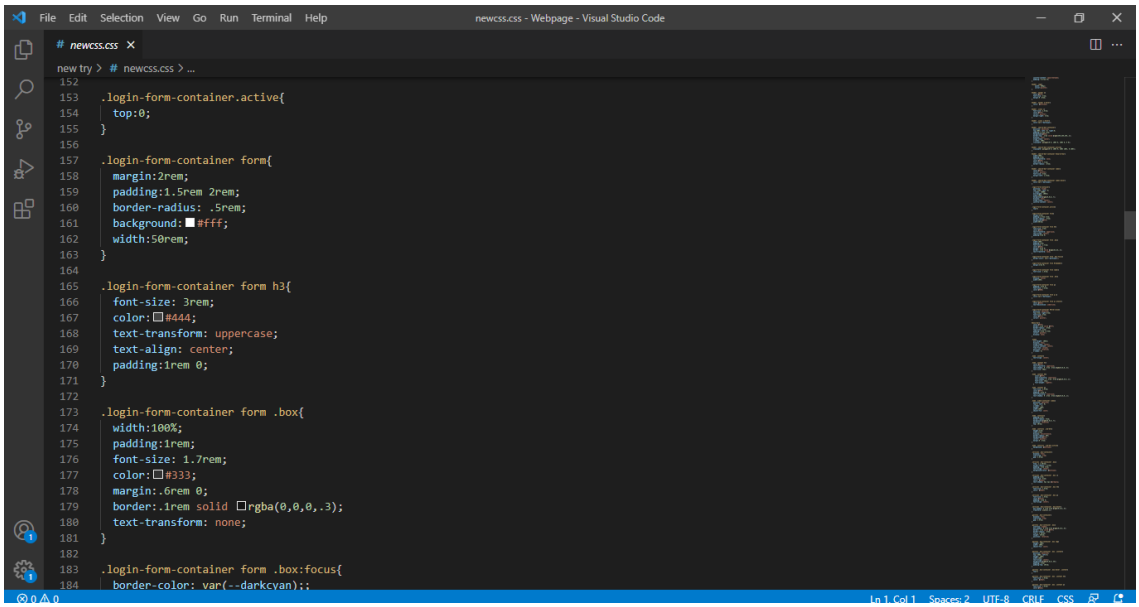


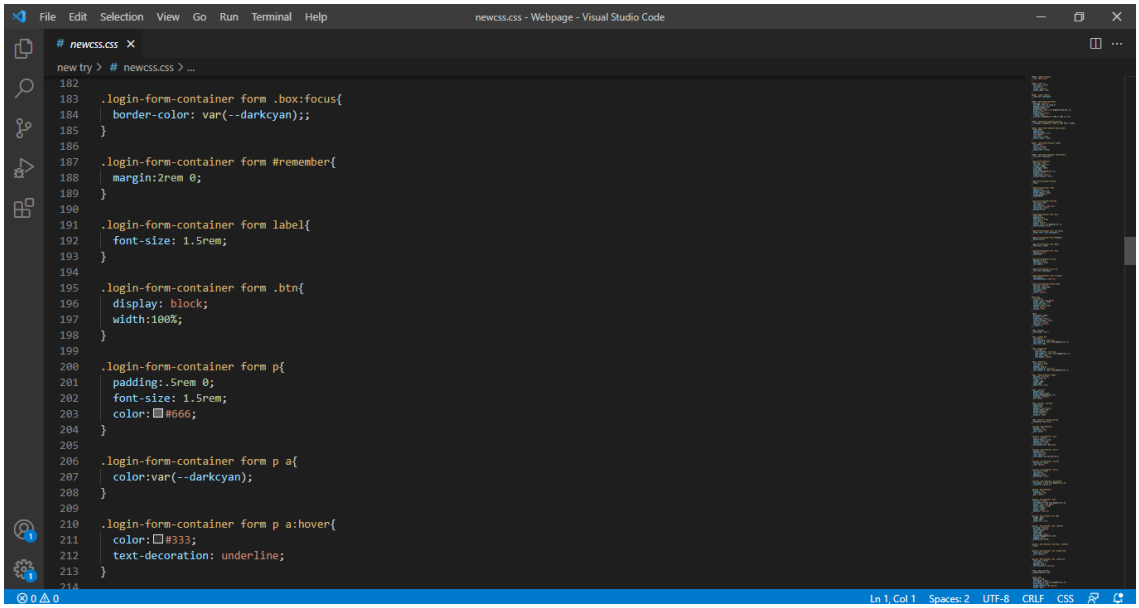


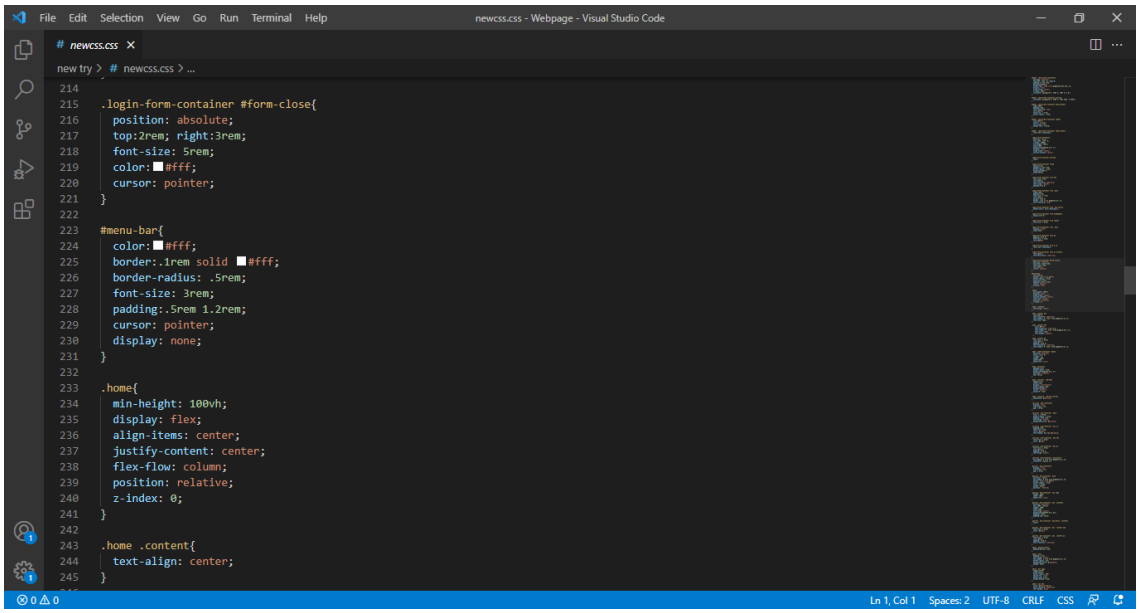

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
button:active {
box-shadow: 2px 5px #666;
transform: translateY(4px);
}
header{
position: fixed;
top:0; left: 0; right:0;
background:#333;
z-index: 1000;
display: flex;
align-items: center;
justify-content: space-between;
padding: 0.5rem 5%;
}
header .logo{
width: 300px;
cursor:pointer;
}
header .navbar a{
color:#fff;
font-size: 2rem;
margin:0 .8rem;
}
header .navbar a:hover{
color: darkcyan;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

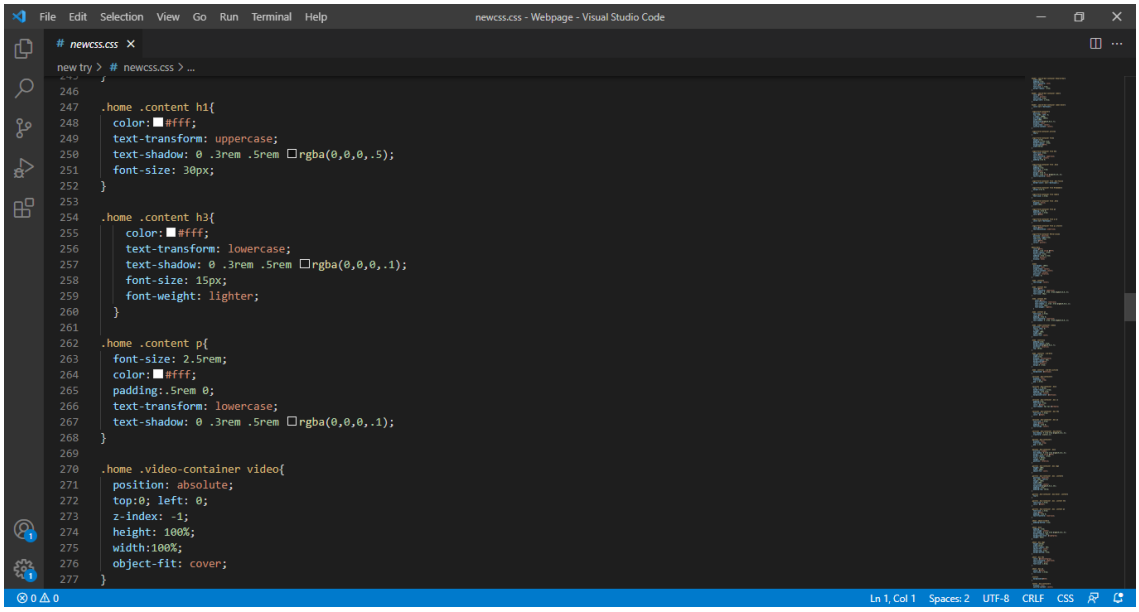

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
header .icons i{
font-size: 2.5rem;
color:#fff;
cursor: pointer;
margin-right: 2rem;
}
header .icons i:hover{
color:var(--darkcyan);
}
header .search-bar-container{
position: absolute;
top:100%; left: 0; right:0;
padding:1.5rem 2rem;
background:#333;
border-top: .1rem solid rgba(255,255,255,.2);
display: flex;
align-items: center;
z-index: 1001;
clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
}
header .search-bar-container.active{
clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);
}
header .search-bar-container #search-bar{
width:100%;
padding:1rem;
text-transform: none;
color:#333;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

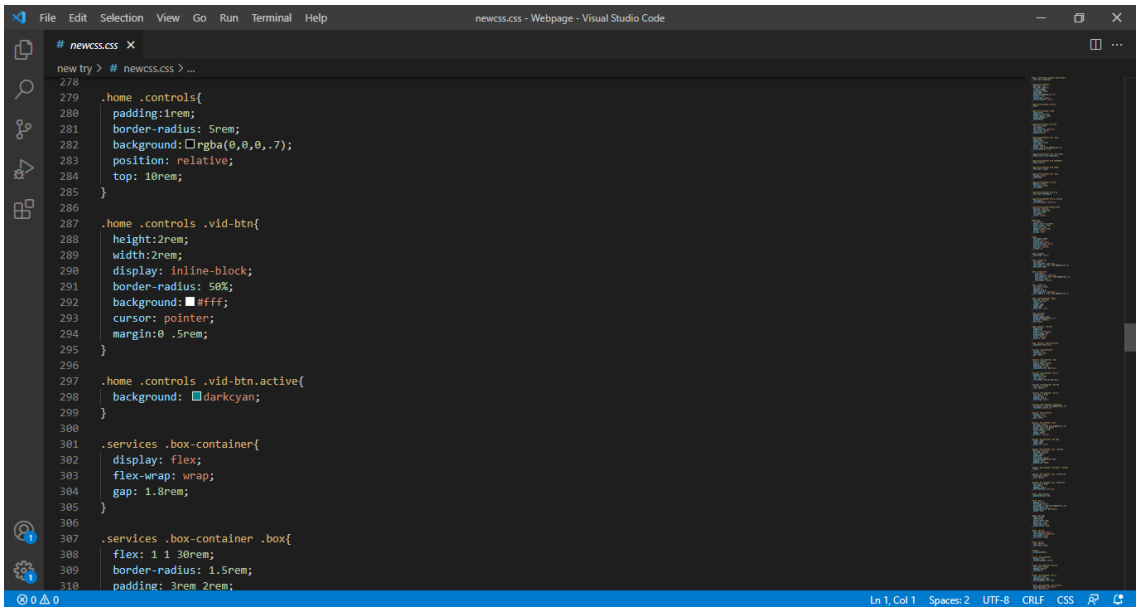

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
header .search-bar-container #search-bar{
width:100%;
padding:1rem;
text-transform: none;
color:#333;
font-size: 1.7rem;
border-radius: .5rem;
}
header .search-bar-container label{
color:#fff;
cursor: pointer;
font-size: 3rem;
margin-left: 1.5rem;
}
header .search-bar-container label:hover{
color:var(--darkcyan);
}
.login-form-container{
position: fixed;
top:-120%; left: 0;
z-index: 10000;
min-height: 100vh;
width:100%;
background:rgba(0,0,0,.7);
display: flex;
align-items: center;
justify-content: center;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

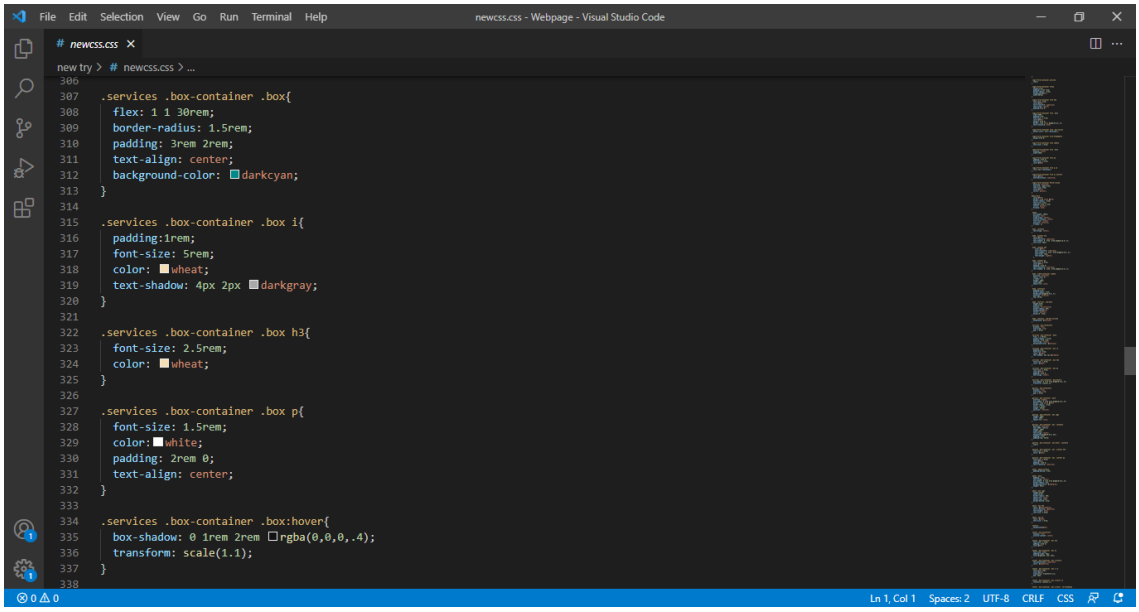

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.login-form-container.active{
top:0;
}
.login-form-container form{
margin:2rem;
padding:1.5rem 2rem;
border-radius: .5rem;
background:#fff;
width:50rem;
}
.login-form-container form h3{
font-size: 3rem;
color:#444;
text-transform: uppercase;
text-align: center;
padding:1rem 0;
}
.login-form-container form .box{
width:100%;
padding:1rem;
font-size: 1.7rem;
color:#333;
margin:.6rem 0;
border:.1rem solid rgba(0,0,0,.3);
text-transform: none;
}
.login-form-container form .box:focus{
border-color: var(--darkcyan);;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

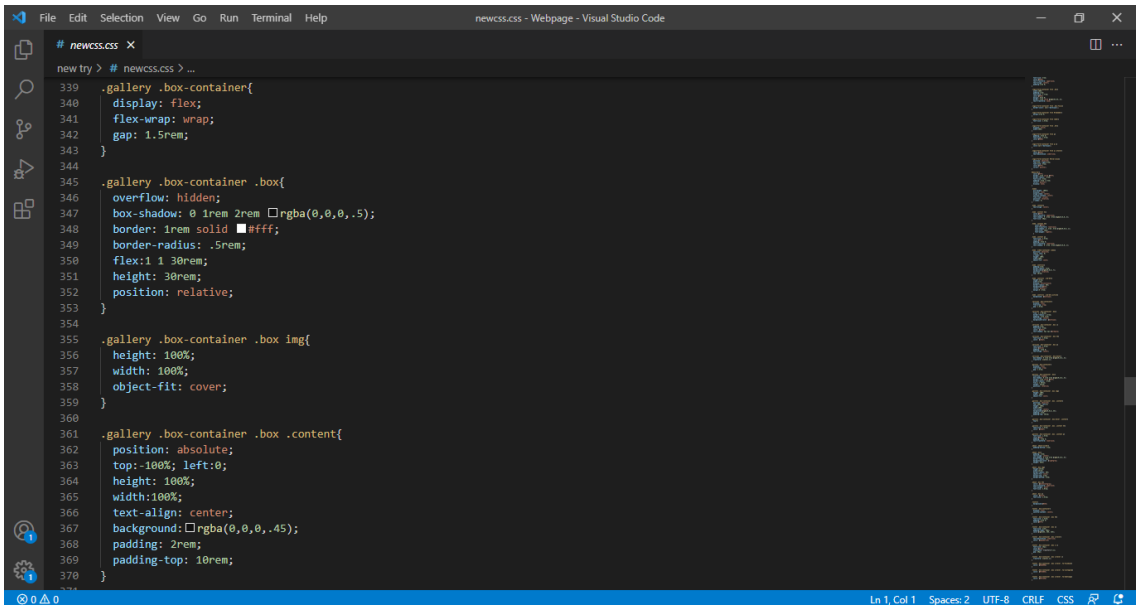

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.login-form-container form .box:focus{
border-color: var(--darkcyan);;
}
.login-form-container form #remember{
margin:2rem 0;
}
.login-form-container form label{
font-size: 1.5rem;
}
.login-form-container form .btn{
display: block;
width:100%;
}
.login-form-container form p{
padding:.5rem 0;
font-size: 1.5rem;
color:#666;
}
.login-form-container form p a{
color:var(--darkcyan);
}
.login-form-container form p a:hover{
color:#333;
text-decoration: underline;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

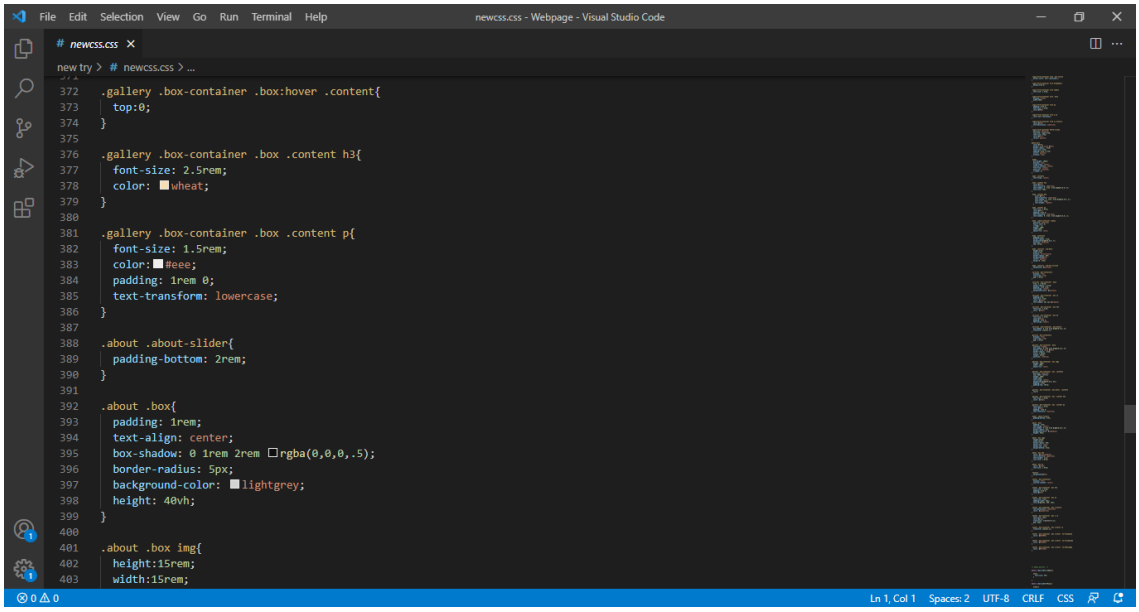

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.login-form-container #form-close{
position: absolute;
top:2rem; right:3rem;
font-size: 5rem;
color:#fff;
cursor: pointer;
}
#menu-bar{
color:#fff;
border:.1rem solid #fff;
border-radius: .5rem;
font-size: 3rem;
padding:.5rem 1.2rem;
cursor: pointer;
display: none;
}
.home{
min-height: 100vh;
display: flex;
align-items: center;
justify-content: center;
flex-flow: column;
position: relative;
z-index: 0;
}
.home .content{
text-align: center;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

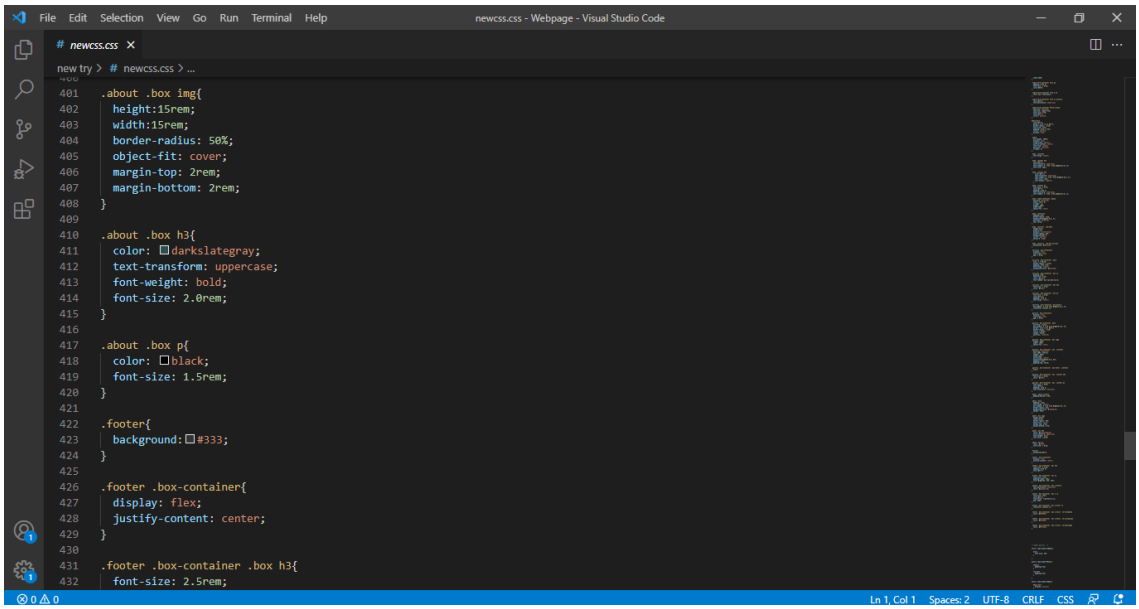

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.home .content h1{
color: #fff;
text-transform: uppercase;
text-shadow: 0 .3rem .5rem rgba(0,0,0,.5);
font-size: 30px;
}
.home .content h3{
color: #fff;
text-transform: lowercase;
text-shadow: 0 .3rem .5rem rgba(0,0,0,.1);
font-size: 15px;
font-weight: lighter;
}
.home .content p{
font-size: 2.5rem;
color: #fff;
padding:.5rem 0;
text-transform: lowercase;
text-shadow: 0 .3rem .5rem rgba(0,0,0,.1);
}
.home .video-container video{
position: absolute;
top:0; left: 0;
z-index: -1;
height: 100%;
width:100%;
object-fit: cover;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

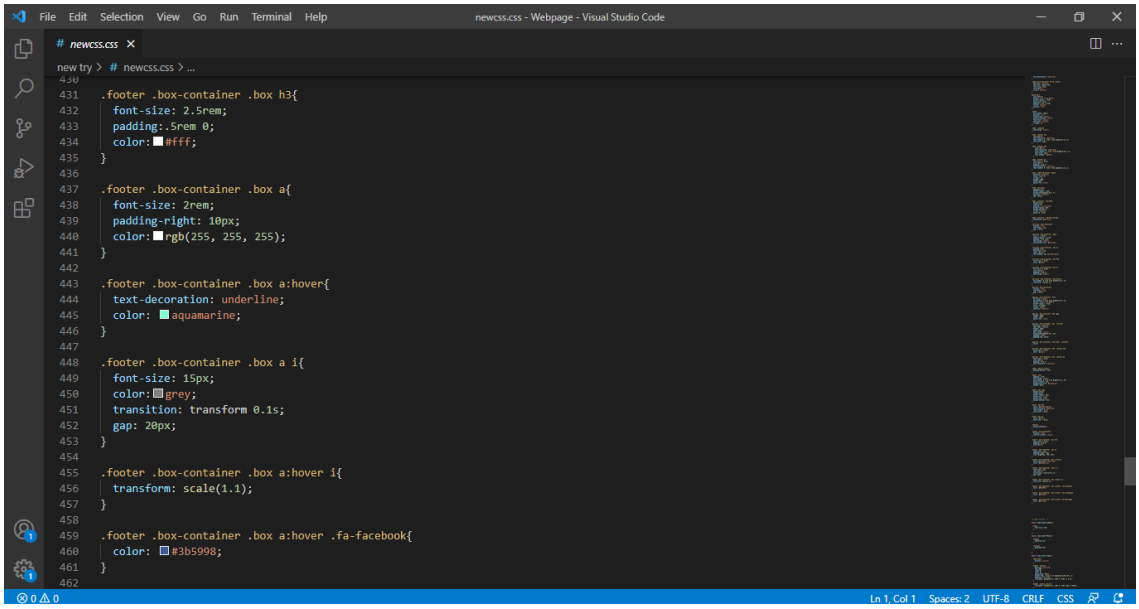

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.home .controls{
padding:1rem;
border-radius: 5rem;
background:rgba(0,0,0,.7);
position: relative;
top: 10rem;
}
.home .controls .vid-btn{
height:2rem;
width:2rem;
display: inline-block;
border-radius: 50%;
background: #fff;
cursor: pointer;
margin:0 .5rem;
}
.home .controls .vid-btn.active{
background: darkcyan;
}
.services .box-container{
display: flex;
flex-wrap: wrap;
gap: 1.8rem;
}
.services .box-container .box{
flex: 1 1 30rem;
border-radius: 1.5rem;
padding: 3rem 2rem;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

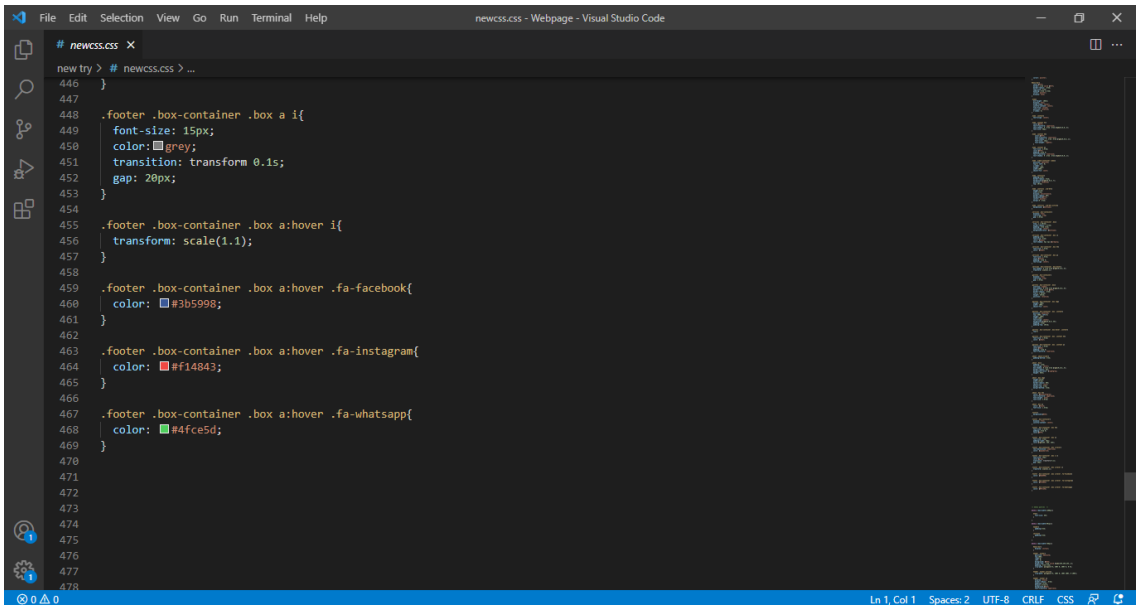

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
306
307 .services .box-container .box{
308 flex: 1 1 30rem;
309 border-radius: 1.5rem;
310 padding: 3rem 2rem;
311 text-align: center;
312 background-color: darkcyan;
313 }
314
315 .services .box-container .box i{
316 padding:1rem;
317 font-size: 5rem;
318 color: wheat;
319 text-shadow: 4px 2px darkgray;
320 }
321
322 .services .box-container .box h3{
323 font-size: 2.5rem;
324 color: wheat;
325 }
326
327 .services .box-container .box p{
328 font-size: 1.5rem;
329 color: white;
330 padding: 2rem 0;
331 text-align: center;
332 }
333
334 .services .box-container .box:hover{
335 box-shadow: 0 1rem 2rem rgba(0,0,0,.4);
336 transform: scale(1.1);
337 }
338
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

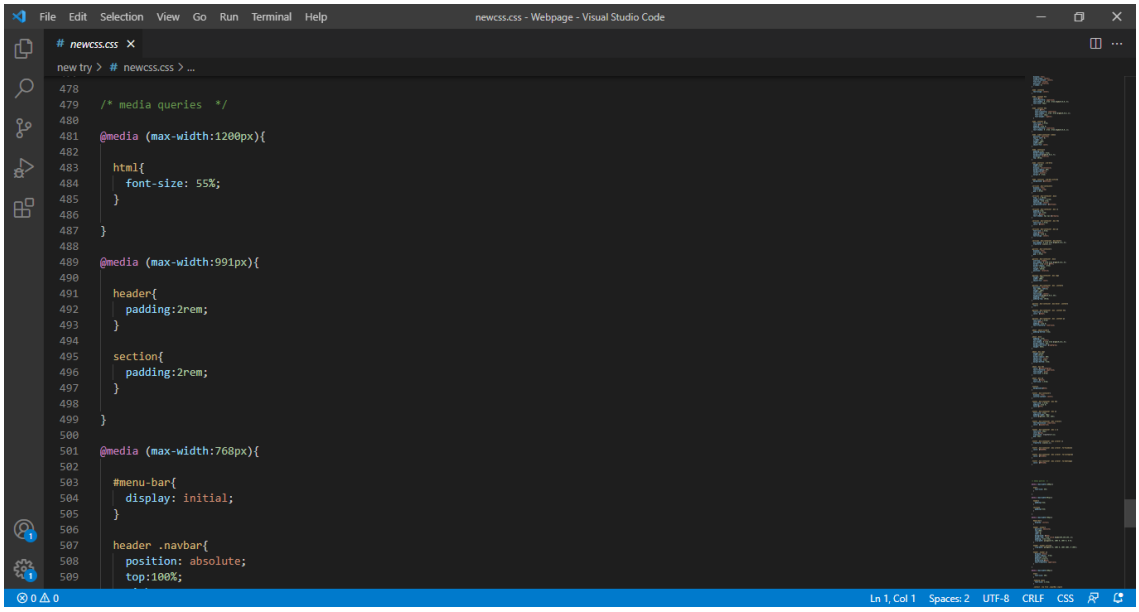

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
339 .gallery .box-container{
340 display: flex;
341 flex-wrap: wrap;
342 gap: 1.5rem;
343 }
344
345 .gallery .box-container .box{
346 overflow: hidden;
347 box-shadow: 0 1rem 2rem rgba(0,0,0,.5);
348 border: 1rem solid #fff;
349 border-radius: .5rem;
350 flex:1 1 30rem;
351 height: 30rem;
352 position: relative;
353 }
354
355 .gallery .box-container .box img{
356 height: 100%;
357 width: 100%;
358 object-fit: cover;
359 }
360
361 .gallery .box-container .box .content{
362 position: absolute;
363 top:-100%; left:0;
364 height: 100%;
365 width:100%;
366 text-align: center;
367 background:rgba(0,0,0,.45);
368 padding: 2rem;
369 padding-top: 10rem;
370 }
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

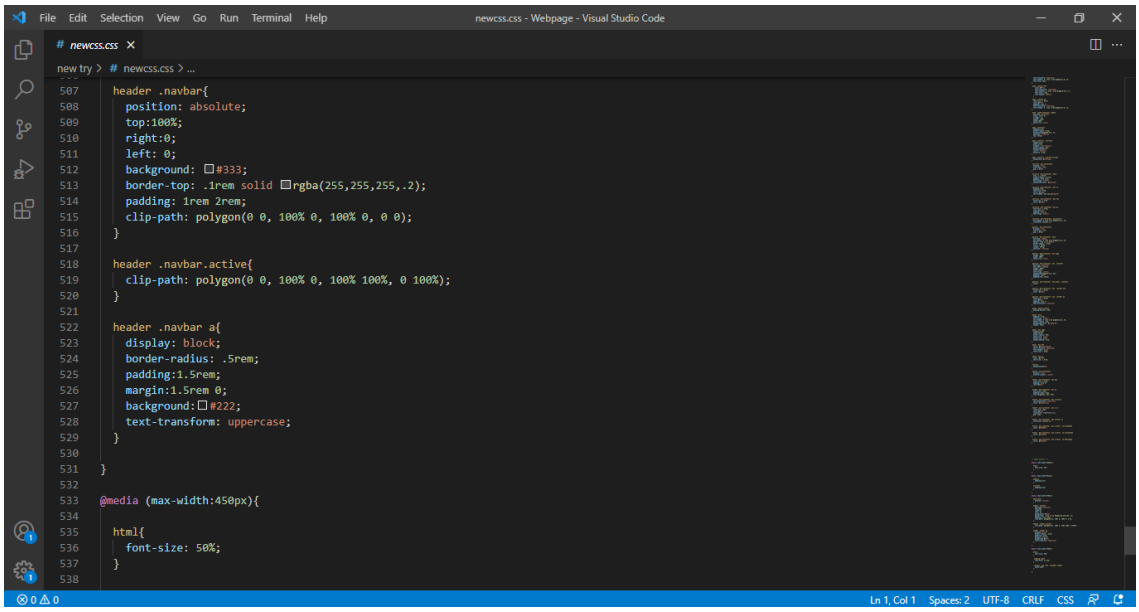

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.gallery .box-container .box:hover .content{
top:0;
}
.gallery .box-container .box .content h3{
font-size: 2.5rem;
color: wheat;
}
.gallery .box-container .box .content p{
font-size: 1.5rem;
color:#eee;
padding: 1rem 0;
text-transform: lowercase;
}
.about .about-slider{
padding-bottom: 2rem;
}
.about .box{
padding: 1rem;
text-align: center;
box-shadow: 0 1rem 2rem rgba(0,0,0,.5);
border-radius: 5px;
background-color: lightgrey;
height: 40vh;
}
.about .box img{
height:15rem;
width:15rem;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

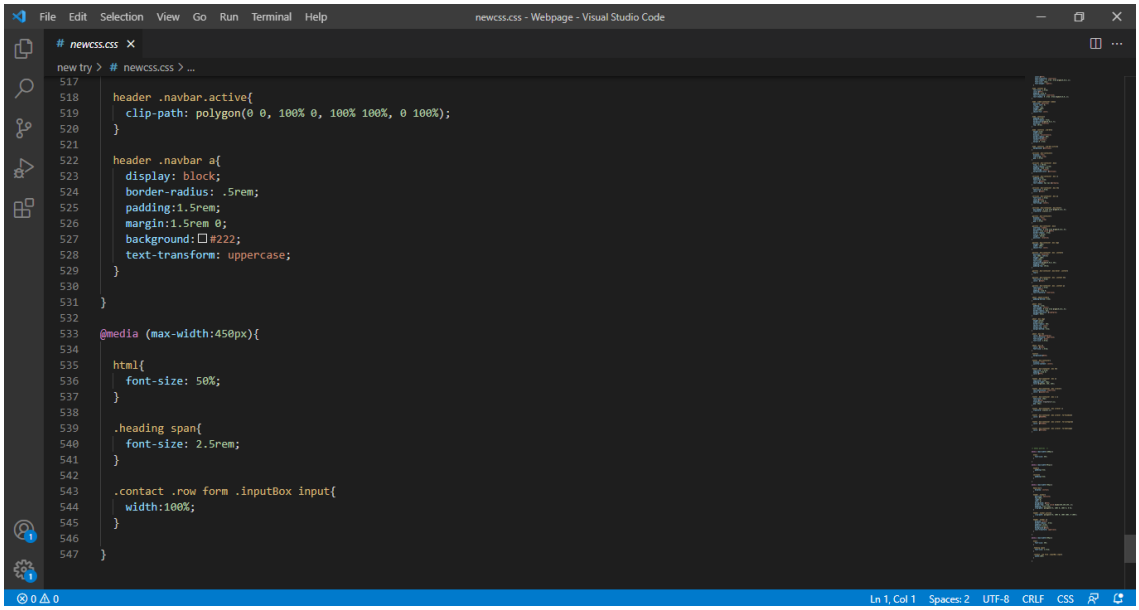

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.about .box img{
height:15rem;
width:15rem;
border-radius: 50%;
object-fit: cover;
margin-top: 2rem;
margin-bottom: 2rem;
}
.about .box h3{
color: darkslategray;
text-transform: uppercase;
font-weight: bold;
font-size: 2.0rem;
}
.about .box p{
color: black;
font-size: 1.5rem;
}
.footer{
background:#333;
}
.footer .box-container{
display: flex;
justify-content: center;
}
.footer .box-container .box h3{
font-size: 2.5rem;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS


File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
.footer .box-container .box h3{
font-size: 2.5rem;
padding:.5rem 0;
color:#fff;
}
.footer .box-container .box a{
font-size: 2rem;
padding-right: 10px;
color:rgb(255, 255, 255);
}
.footer .box-container .box a:hover{
text-decoration: underline;
color: aquamarine;
}
.footer .box-container .box a i{
font-size: 15px;
color:grey;
transition: transform 0.1s;
gap: 20px;
}
.footer .box-container .box a:hover i{
transform: scale(1.1);
}
.footer .box-container .box a:hover .fa-facebook{
color: #3b5998;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

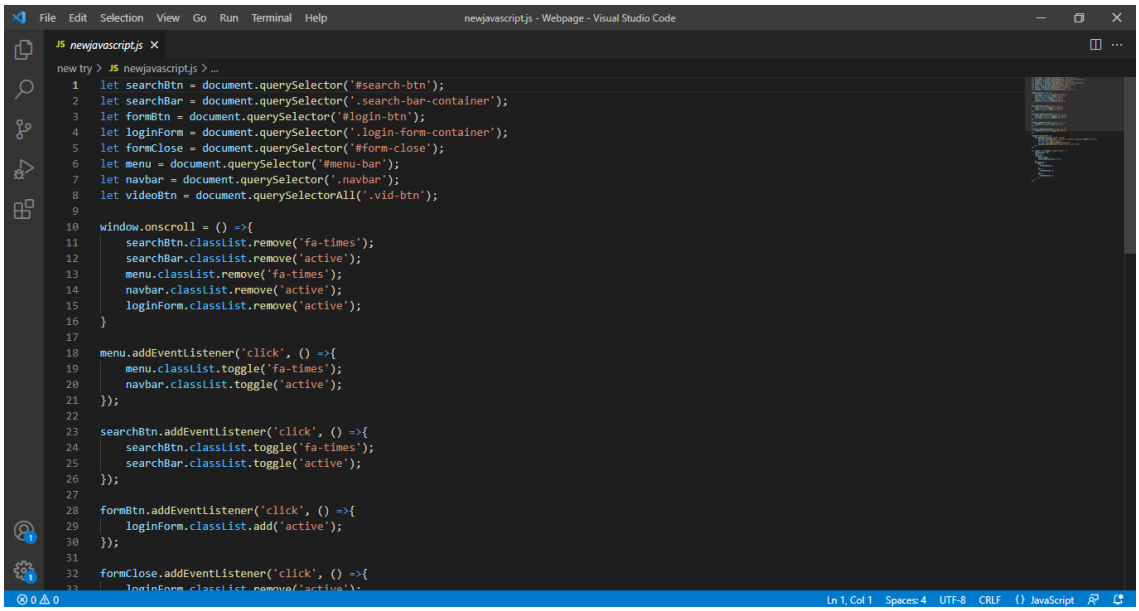

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
}
.footer .box-container .box a i{
font-size: 15px;
color:grey;
transition: transform 0.1s;
gap: 20px;
}
.footer .box-container .box a:hover i{
transform: scale(1.1);
}
.footer .box-container .box a:hover .fa-facebook{
color: #3b5998;
}
.footer .box-container .box a:hover .fa-instagram{
color: #f14843;
}
.footer .box-container .box a:hover .fa-whatsapp{
color: #4fce5d;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS


File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
/* media queries */
@media (max-width:1200px){
html{
font-size: 55%;
}
}
@media (max-width:991px){
header{
padding:2rem;
}
section{
padding:2rem;
}
}
@media (max-width:768px){
#menu-bar{
display: initial;
}
header .navbar{
position: absolute;
top:100%;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

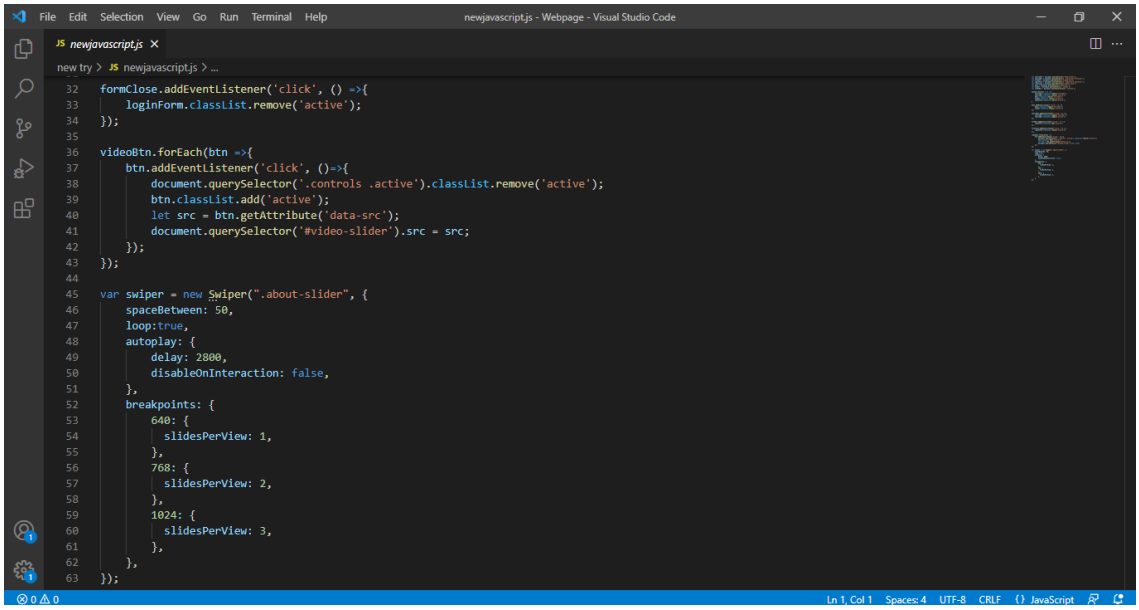

File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > newcss.css > ...
header .navbar{
position: absolute;
top:100%;
right:0;
left: 0;
background: #333;
border-top: .1rem solid rgba(255,255,255,.2);
padding: 1rem 2rem;
clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
}
header .navbar.active{
clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);
}
header .navbar a{
display: block;
border-radius: .5rem;
padding:1.5rem;
margin:1.5rem 0;
background: #222;
text-transform: uppercase;
}
}
@media (max-width:450px){
html{
font-size: 50%;
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS


File Edit Selection View Go Run Terminal Help
newcss.css - Webpage - Visual Studio Code
newcss.css
new try > # newcss.css > ...
header .navbar.active{
clip-path: polygon(0 0, 100% 0, 100% 100%, 0 100%);
}
header .navbar a{
display: block;
border-radius: .5rem;
padding:1.5rem;
margin:1.5rem 0;
background: #222;
text-transform: uppercase;
}
}
@media (max-width:450px){
html{
font-size: 50%;
}
.heading span{
font-size: 2.5rem;
}
.contact .row form .inputBox input{
width:100%;
}
}
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS

Appendix D : JavaScript Coding



```
let searchBtn = document.querySelector('#search-btn');
let searchBar = document.querySelector('.search-bar-container');
let formBtn = document.querySelector('#login-btn');
let loginForm = document.querySelector('.login-form-container');
let formClose = document.querySelector('#form-close');
let menu = document.querySelector('#menu-bar');
let navbar = document.querySelector('.navbar');
let videoBtn = document.querySelectorAll('.vid-btn');

window.onscroll = () =>{
    searchBtn.classList.remove('fa-times');
    searchBar.classList.remove('active');
    menu.classList.remove('fa-times');
    navbar.classList.remove('active');
    loginForm.classList.remove('active');
}

menu.addEventListener('click', () =>{
    menu.classList.toggle('fa-times');
    navbar.classList.toggle('active');
});

searchBtn.addEventListener('click', () =>{
    searchBtn.classList.toggle('fa-times');
    searchBar.classList.toggle('active');
});

formBtn.addEventListener('click', () =>{
    loginForm.classList.add('active');
});

formClose.addEventListener('click', () =>{
```



```
    loginForm.classList.remove('active');
});

videoBtn.forEach(btn =>{
    btn.addEventListener('click', ()=>{
        document.querySelector('.controls .active').classList.remove('active');
        btn.classList.add('active');
        let src = btn.getAttribute('data-src');
        document.querySelector('#video-slider').src = src;
    });
});

var swiper = new Swiper(".about-slider", {
    spaceBetween: 50,
    loop:true,
    autoplay: {
        delay: 2800,
        disableOnInteraction: false,
    },
    breakpoints: {
        640: {
          slidesPerView: 1,
        },
        768: {
          slidesPerView: 2,
        },
        1024: {
          slidesPerView: 3,
        },
    },
});
```