NSBM Green University

Faculty of Computing

<Navigation Unit>

PUSL2021 & Computing Group Project

Lecturer: Pramudya thilakarathne

Content

Introduction

Our project aims to address the issue of guests struggling to navigate around the campus of the National School of Business Management (NSBM) in Sri Lanka. We are proposing to develop an innovative IoT device that will enable users to easily locate the faculties, canteens, and departments within the university. The device will provide users with real-time information about their current location, helping them to navigate through the campus with ease. Additionally, our project includes an application that can be used by the university's security personnel to track the movement of guests within the campus, enhancing the overall safety and security of the university. This project has the potential to revolutionize the way in which guests navigate through NSBM and ensure that they have a pleasant experience.

Facts Gathering techniques

- Asking people questions about a specific topic to collect data. They can be conducted in person, over the phone, or online.

- Interviews with seniors: Interviews involve having a one-on-one conversation with some seniors who have more experience about this specific topic. to gather information about their experiences or opinions on a particular subject.

- Observations: We invented this idea by this technique. It’s not something like when we used this technique to gather data, we started seeing that our students have a hard time finding the places.

- Case studies: we have searched some previous projects’ case studies to gather information. We have seen that these studies have helped us to develop the project and to gain more knowledge in this field.
- These are some of them.

□"Indoor Navigation System for Large Buildings using Augmented Reality

□"An Indoor Positioning System for a University Campus using Bluetooth Low Energy Beacons."

□"Design and Implementation of an Indoor Navigation System for a University Building".

Project Objectives

- To develop an IoT device that can accurately identify the location of users within the university.
- To provide users with real-time information about their current location, making it easier for them to navigate around the university.
- To create an application that can be used by the university's security personnel to track the movement of guests within the campus, enhancing safety and security.
- To enhance the experience of guests visiting the university by providing them with an innovative and user-friendly solution for navigation.
- To reduce the time and effort required for guests to find the correct faculties, canteens, and departments within the university.
- To increase the efficiency and effectiveness of security measures within the university.

Overall, our project aims to leverage the power of IoT technology to provide a comprehensive solution for navigation and security within the NSBM campus, ultimately enhancing the experience of guests and improving the overall safety and security of the university.

Problem

The problem that we identified for the project is the difficulty faced by guests in navigating through the NSBM campus. Many guests, especially those visiting the university for the first time, find it challenging to locate the correct faculties, canteens, and departments. This leads to frustration, wasted time, and an overall negative experience. The lack of an efficient navigation system within the campus also poses security risks, as it becomes challenging for security personnel to track the movement of guests accurately. Thus, our project aims to solve this problem by developing an innovative IoT device and application that can accurately identify the location of users and provide real-time information about their current location, making it easier for them to navigate through the campus and enhancing safety and security measures.

Solution

The solution that we identified for the problem of guest navigation within the NSBM campus is the development of an IoT device and an application. The IoT device will use various sensors to accurately identify the location of users within the campus and display the location on the device's interface. The device will also have a user-friendly interface that provides real-time information about the user's current location, enabling them to navigate through the campus with ease.

Additionally, we are developing an application that will be used by security personnel to track the movement of guests within the campus. The application will display the location of all devices being used within the campus, enabling security personnel to respond quickly to any potential security risks.

Together, these solutions aim to provide a comprehensive navigation system within the NSBM campus that will make it easier for guests to find their way around while also enhancing security measures. By leveraging IoT technology, we can provide real-time information that ensures guests have a pleasant and hassle-free experience within the campus.

Methodologies and approaches we used

For our project, we adopted the Agile methodology, which involved a flexible and iterative approach to software development. We followed the following methodologies and approaches:

- Gathered Information: In the initial phase, we gathered information about the problem faced by guests in navigating through the NSBM campus. We conducted research to understand the pain points and challenges of guests.
- Planning (Designing): Once we had gathered sufficient information, we moved onto the planning phase, where we designed the solution to address the identified problem. We created a detailed plan for the development of the IoT device, the application, and the web page.
- Hardware Connecting (Coding Arduino): In the development phase, we started working on the hardware component of the project, which involved coding the Arduino board to connect the sensors and other components to accurately identify the location of users within the campus.
- Web Page Creation: We also developed a web page that could be used to track the movement of guests within the campus. This web page was designed to display the location of all devices being used within the campus and enable security personnel to respond quickly to any potential security risks.
- Testing the Prototype: Once the hardware and software components were developed, we tested the prototype to ensure that it functioned as expected. We conducted various tests and iterations to refine the solution and make it more efficient and effective.

Overall, our approach involved a flexible and iterative process that allowed us to identify the problem, design the solution, develop the hardware and software components, and test the prototype effectively. The Agile methodology helped us to adapt to changes and refine the solution continuously, resulting in a robust and user-friendly solution for guest navigation within the university.

User deliverables

The user deliverables for our project include an IoT unit and a QR code. These deliverables are designed to provide a comprehensive solution for guest navigation and security within the NSBM campus.

1. IoT Unit: The IoT unit is a portable device that users can carry with them as they navigate through the campus. The device is equipped with various sensors that accurately identify the user's location and display it on the device's interface. The interface is user-friendly and provides real-time information about the user's current location, enabling them to navigate through the campus with ease.
2. QR Code: The QR code is designed to be used by security personnel to track the movement of guests within the campus. Each IoT unit is associated with a unique QR code that can be scanned by security personnel to obtain information about the user's location. This information is displayed on a web page that can be accessed by security personnel, enabling them to respond quickly to any potential security risks.

Together, these user deliverables provide a comprehensive solution for guest navigation and security within the university. Users can carry the IoT unit with them as they navigate through the university, while security personnel can use the QR code to track their movement and respond quickly to any potential security risks.

For which users the device will be most valuable for?

This IOT device and QR code solution are made to give visitors to the NSBM university, specially first-time visitors, an easy and effective way to explore the university. First-time visitors frequently have trouble navigating the university since they are not familiar with university layout. Our IOT devices will be especially useful in these circumstances because it gives visitors precise real-time location information to make it easier for them to explore the university.

Visitors can quickly identify their current location and follow their moves throughout the university in real-time by using the capabilities of our IoT device and QR code solution. As a result, visitors will have a better overall experience because they won't need to rely on outdated paper maps or seek for directions from others, which not only saves time and effort but also improves the entire tourist experience.

Overall, this IOT device and QR code solution provides first-time visitors to the NSBM university with a convenient and efficient way to navigate through the university, making their visit more pleasurable.

Additionally, the solution will be valuable for security personnel responsible for monitoring the campus. The QR code associated with each IoT device will enable them to track the movement of guests within the campus, helping them to identify any potential security risks and respond quickly to them.

Testing and Evaluation

For testing and evaluating our IoT device, we conducted various tests to ensure that it functioned as expected and met our project objectives. One of the test cases that we performed involved checking the accuracy of the device's location tracking near the faculty of computing.

The test case involved the following steps:

1. We placed the IoT device near the faculty of computing, where there are many buildings and different routes.

2. We turned on the device and waited for it to connect to the sensors and other components.

3. We checked the device's interface to ensure that it displayed the correct location of the device.

4. We walked around the faculty of computing with the device and checked the interface again to ensure that the device's location was upda

5. We repeated the test several times, taking different routes and paths, to ensure that the device accurately tracked the user's location.

6. We noted any inaccuracies or errors in the device's location tracking and made changes to the device's software and hardware components to improve its accuracy.

After conducting the test case, we found that the IoT device accurately tracked the user's location near the faculty of computing. The device's interface displayed the correct location of the device in real-time, and we did not encounter any significant inaccuracies or errors in the device's location tracking.

Test Case ID	Test Case Name	Test case objective	Expected Output	Test case steps	Test Data	Expected results	Post Condition	Actual Result	Status (Pass/ Fail)
1	Location Display Test	To verify that the navigation unit correctly displays the location name on the display	The navigation unit displays the correct location name on the display and updates it when the user moves to a new location.	1) Power On the navigation (IOT) unit. 2) Wait for the device to acquire GPS signal and display the current location. 3) Verify that the displayed location and the actual location of the user match. 4) Move to another location and wait for the device to update the displayed location. 5) Verify that the updated location is correct	Current Location: NSBM Green University, Homagama , Sri Lanka Expected location name displayed on the navigation unit: “NSBM Green University”. Update Location: NSBM City Campus, Colombo, Sri Lanka.	The navigation unit displays the correct location name on the display and updates it when the user moves to a new location.	Navigation unit correctly displays the updated location name on the display and continues to update the location in real-time as the user moves to new locations.	Same as we Expected	Pass
2	Location Tracking Test	To verify that the navigation unit tracks the user's location accurately and updates it in real-time	The navigation unit accurately tracks the user's location and updates it in real-time, even in areas with weak GPS signal	1) Switch on the navigation unit. 2) Wait for the device to acquire GPS signal and display the current location. 3) Move to another location and observe the device's ability to track the user's movement. 4) Verify that the displayed location updates in real-time as the user moves. 5) Test the device in different locations with varying GPS signal strengths and observe its ability to track the user's location accurately	Various location within the NSBM campus with different GPS signal strengths. A device with a GPS module and display the user’s location. A user to move around the campus to simulate different scenarios.	The navigation unit accurately tracks the user's location and updates it in real-time, even in areas with weak GPS signal	Navigation unit accurately tracks the user’s location in real-time, even in areas with weak GPS signal. The device should display the correct location and update it as the user moves, without any errors or malfunctions	Same as we Expected	Pass

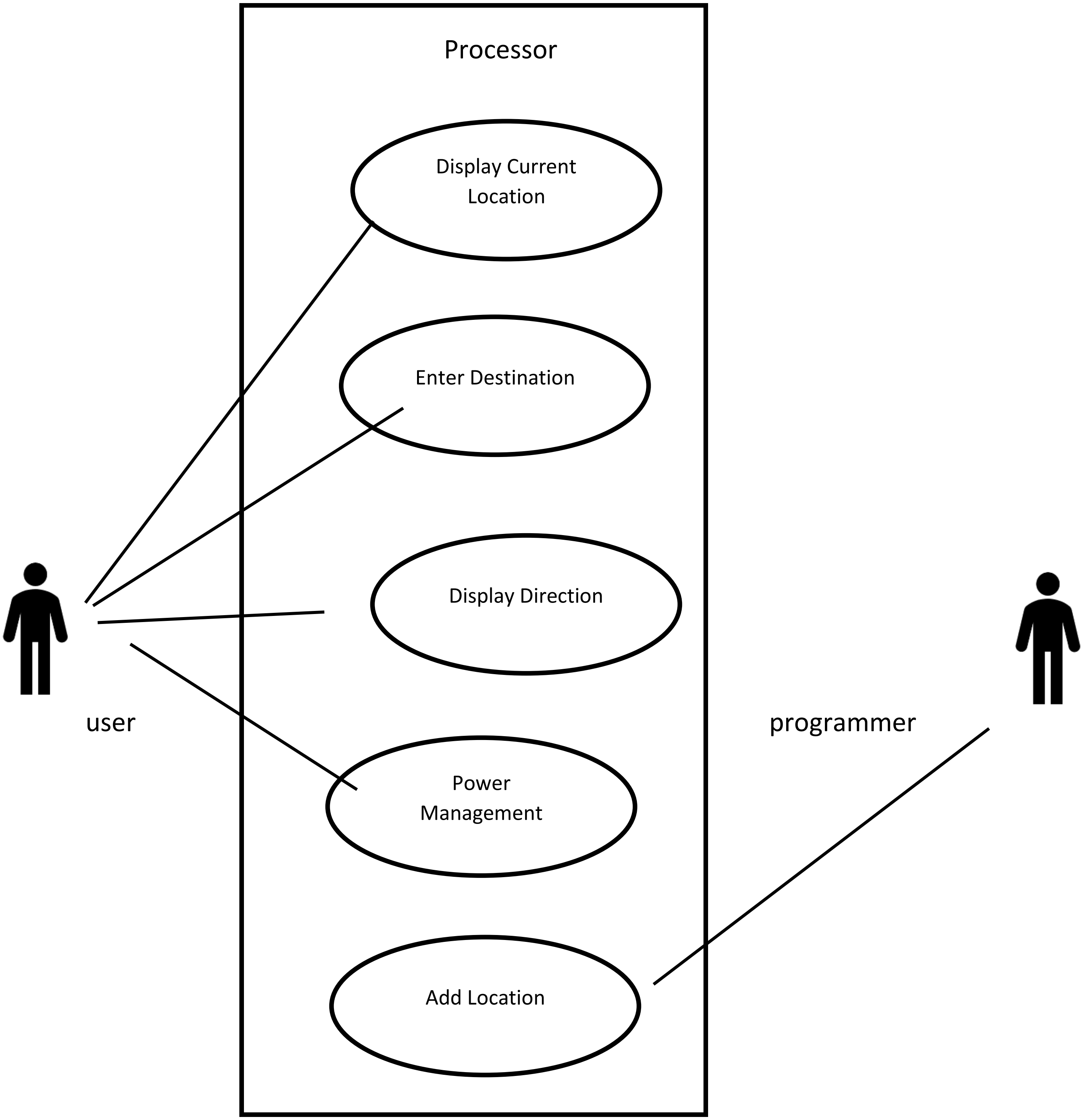
Future Enhancements

There are several future enhancements that could be made to our IoT device and QR code solution. These include:

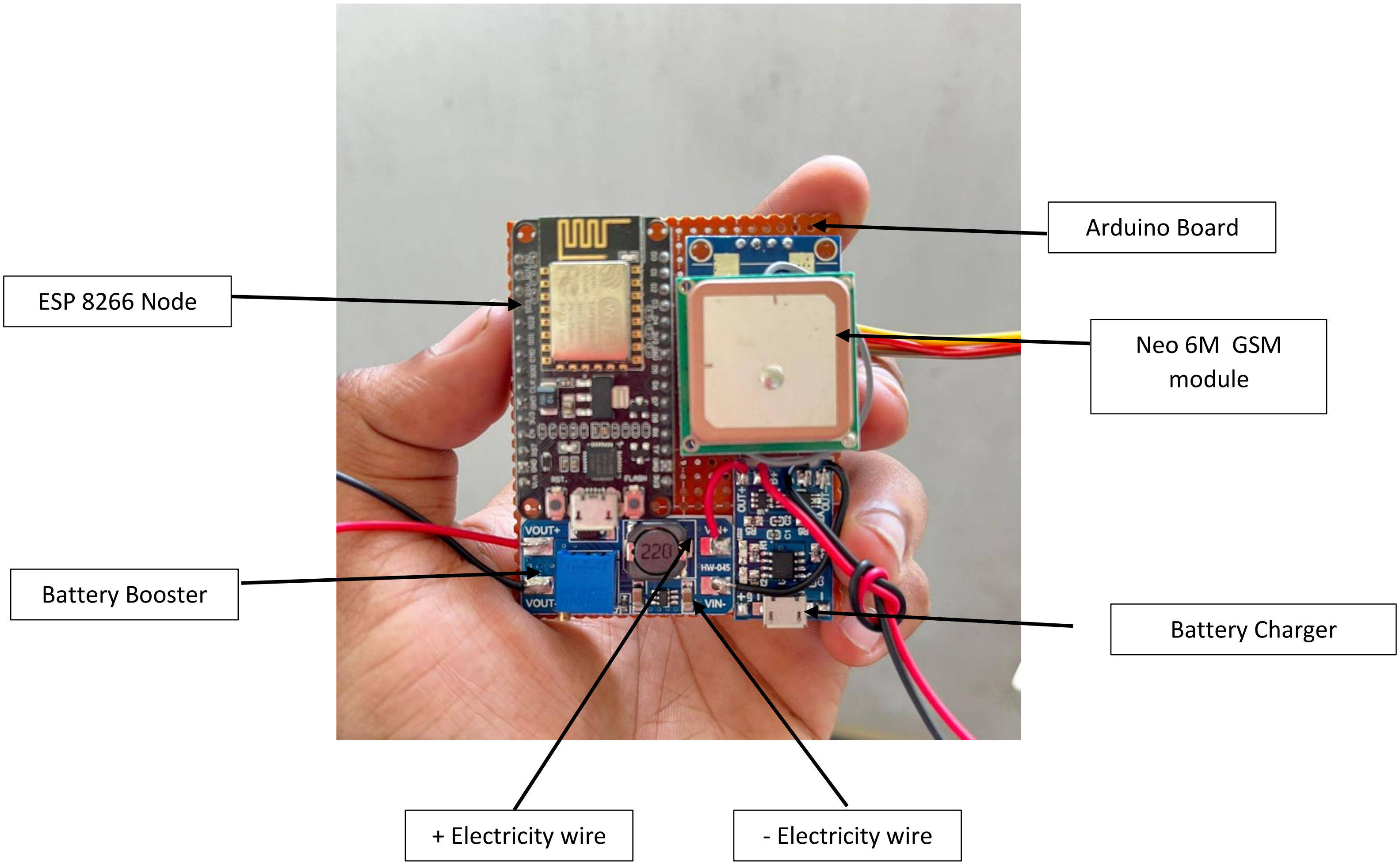
1. AR Technology: Using a 360-degree camera to capture images of the interior of buildings, creating a 3D model of the interior, and incorporating the 3D model into the navigation unit's web page using AR technology. This would allow users to see the 3D model superimposed on their real-world view through their mobile device's camera.
2. Smaller Size: Using surface-mount technology (SMT) components, a smaller GPS module and display, a more efficient power management system, and a smaller and more efficient microcontroller to create a more portable and user-friendly navigation unit that could be easily carried in a pocket or backpack.
3. Touch Screen Display: Adding a touch screen display to the device to guide the user to their desired location. This would allow users to interact with the device more easily and receive more detailed information about the campus.

   It is Much more expensive and it will take much more time and resources to us to make such a developed product .So as the future improvement of this project team suggest to make the smart badge as small as possible it will make the product more user friendly.

.


Processor
Display Current Location
Enter Destination
Display Direction
Power Management
Add Location
user
programmer

Devices used



Inside Process OF this unit

Inside Process:

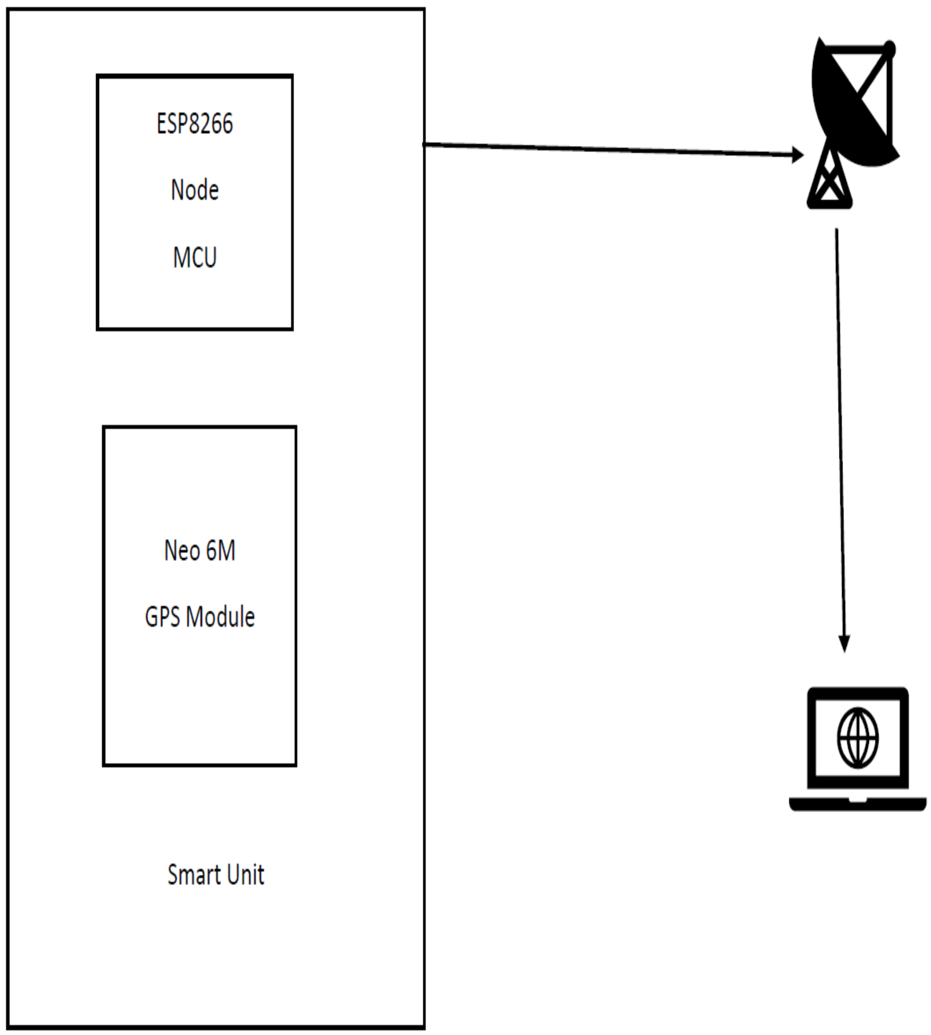
Our navigation unit consists of several components that work together to provide accurate and up-to-date directions to users. The Node 6M GPS module communicates with the ESP8266 microcontroller using the serial communication protocol to send location data. The ESP8266 then processes this data and sends it to the display to show the current location name.

The power generator charges the battery using kinetic energy, while the power booster ensures that the ESP8266 receives a stable power supply. The QR code is generated using a QR code generator tool and links to the web page hosted on the ESP8266**.**

In here this unit will displayed the location name in the display

1. ESP8266 microcontroller: It is a low-cost Wi-Fi-enabled microcontroller with integrated TCP/IP protocol stack that can be used to connect IoT devices to the internet.
2. Node 6M GPS module: It is a GPS module that can receive signals from GPS satellites and calculate the current location based on the received signals.
3. Arduino board: It is an open-source electronics platform based on easy-to-use hardware and software that can be used to build a wide range of IoT projects.
4. Display: It is used to show the current location name and the directions to the desired location. The display is powered by a 3.7V battery.
5. Power generator: It is used to charge the battery that powers the display and other components. The power generator can be either solar or kinetic.
6. Power booster: It ensures that the ESP8266 microcontroller receives a stable power supply.
7. QR code: It connects to the web page hosted within the IoT unit. When scanned with a mobile device, it redirects to the web page that displays the current location and allows the user to enter a desired location to get directions.

Structure Of the System



Hardware requirements

Certainly! The ESP8266 microcontroller is a small and low-cost device that can connect to the internet via Wi-Fi. It has an integrated TCP/IP protocol stack which makes it easy to connect to other IoT devices or web services. This device is used in our navigation unit to connect to the internet and retrieve data about the user's current location and the directions to a desired location.

The Node 6M GPS module is a specialized device used to determine the current location of the user. It works by receiving signals from GPS satellites and calculating the user's latitude and longitude based on the signals received. This module is crucial to the functionality of our navigation unit as it allows us to accurately determine the user's current location and provide accurate directions to a desired location.

The Arduino board is an open-source electronics platform that provides an easy-to-use hardware and software interface for building IoT projects. In our navigation unit, the Arduino board is used to process data received from the GPS module and send it to the ESP8266 microcontroller for transmission over Wi-Fi. The Arduino board also controls the display and manages the power system for the unit.

The display is an essential component of our navigation unit as it provides the user with information about their current location and the directions to a desired location. The display is powered by a 3.7V battery and is controlled by the Arduino board.

The power generator is used to charge the battery that powers the display and other components of the navigation unit. There are several types of power generators that can be used, including solar and kinetic. The choice of power generator depends on the specific needs of the project.

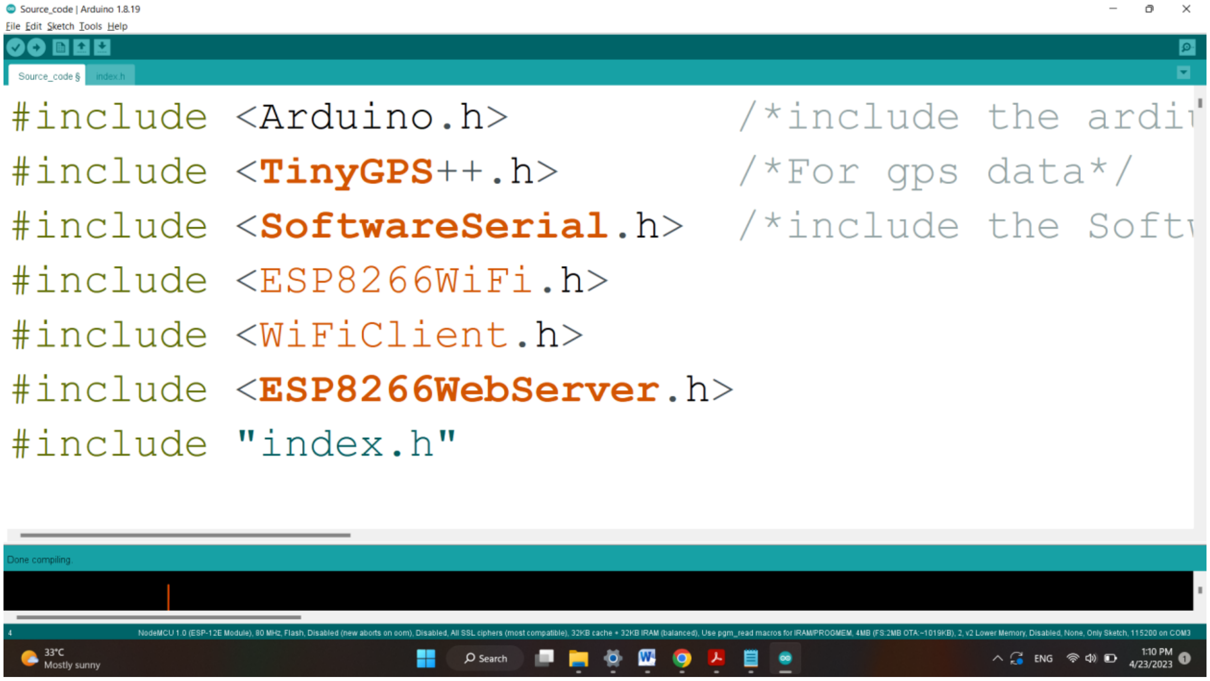
The power booster is a device used to ensure that the ESP8266 microcontroller receives a stable power supply. This is important because unstable power can cause the device to malfunction or even be damaged.

The QR code is a special type of barcode that can be scanned with a mobile device to connect to a web page. In our navigation unit, the QR code connects to a web page hosted within the IoT unit. When scanned with a mobile device, it redirects the user to the web page that displays their current location and allows them to enter a desired location to get directions.

To connect all these devices, we first connected the GPS module to the Arduino board using wires. We then connected the display and power system to the Arduino board. Next, we connected the ESP8266 microcontroller to the Arduino board using serial communication. Finally, we generated a QR code and linked it to the web page hosted within the IoT unit. With all these components connected and working together, we were able to build a functional navigation unit that can accurately determine the user's location and provide directions to a desired location.

Appendices

These are the codes of arduino and web page

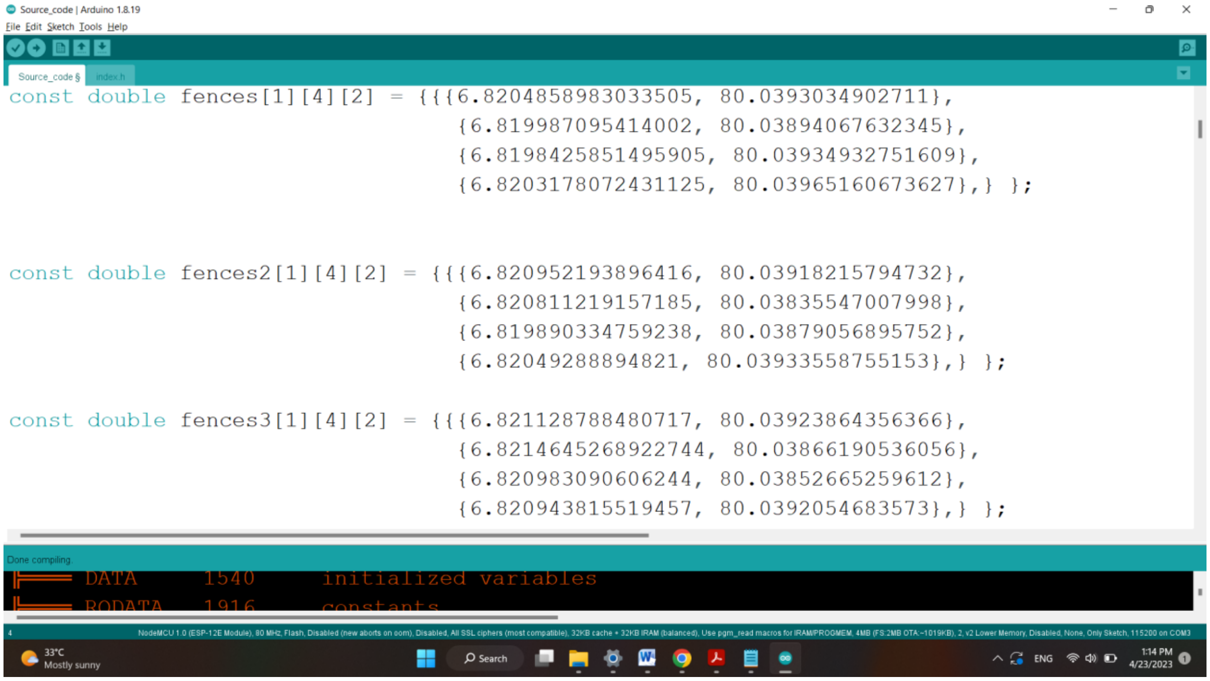


```
#include <Arduino.h>          /*include the ardu
#include <TinyGPS++.h>        /*For gps data*/
#include <SoftwareSerial.h>   /*include the Softw
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include "index.h"
```

```
Done compiling.
```

```
4    NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
```

In here we included the necessary libraries for our function as needed.



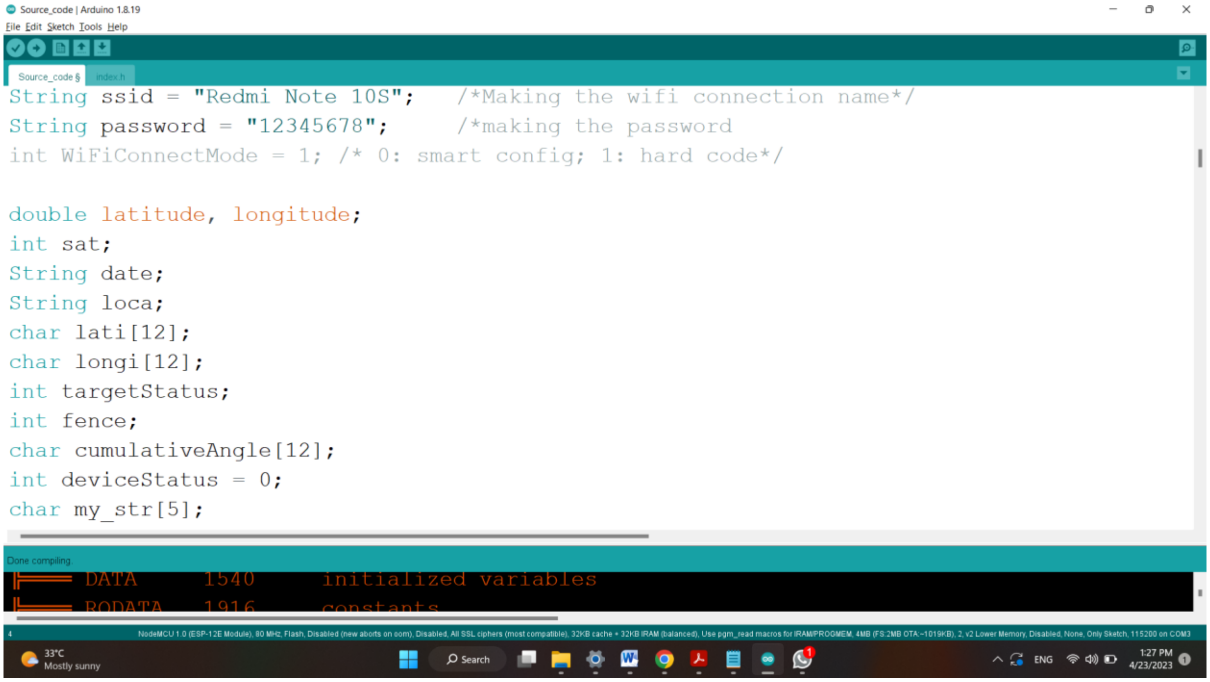
```
const double fences[1][4][2] = {{{6.8204858983033505, 80.0393034902711},
                                  {6.819987095414002, 80.03894067632345},
                                  {6.8198425851495905, 80.03934932751609},
                                  {6.8203178072431125, 80.03965160673627},} };


const double fences2[1][4][2] = {{{6.820952193896416, 80.03918215794732},
                                  {6.820811219157185, 80.03835547007998},
                                  {6.819890334759238, 80.03879056895752},
                                  {6.82049288894821, 80.03933558755153},} };

const double fences3[1][4][2] = {{{6.821128788480717, 80.03923864356366},
                                  {6.8214645268922744, 80.03866190536056},
                                  {6.820983090606244, 80.03852665259612},
                                  {6.820943815519457, 80.0392054683573},} };
```

Done compiling.

```
╠══ DATA      1540     initialized variables
╚══ RODATA    1916     constants
```

4 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

In here we are storing the latitude and longitudes in the programme.



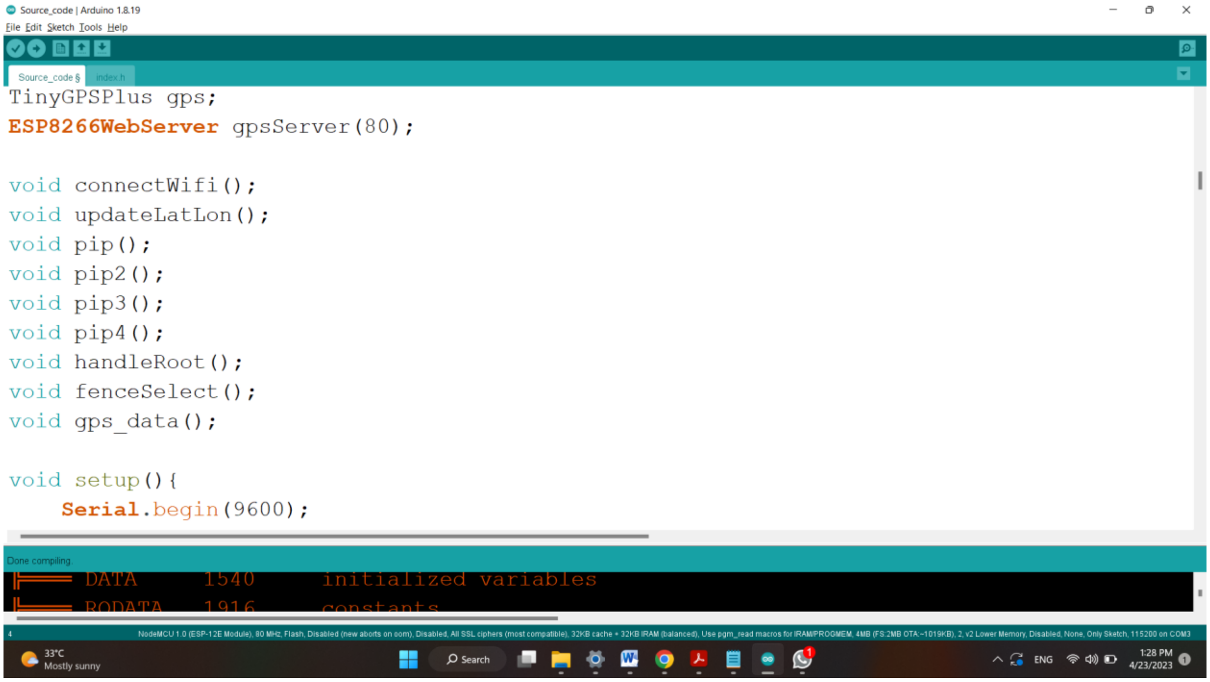
```
String ssid = "Redmi Note 10S";     /*Making the wifi connection name*/
String password = "12345678";       /*making the password
int WiFiConnectMode = 1; /* 0: smart config; 1: hard code*/

double latitude, longitude;
int sat;
String date;
String loca;
char lati[12];
char longi[12];
int targetStatus;
int fence;
char cumulativeAngle[12];
int deviceStatus = 0;
char my_str[5];
```

Done compiling.

```
╠══ DATA      1540     initialized variables
╚══ RODATA    1916     constants
```

4 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

Initializing the Connection SSID and Connection Password.



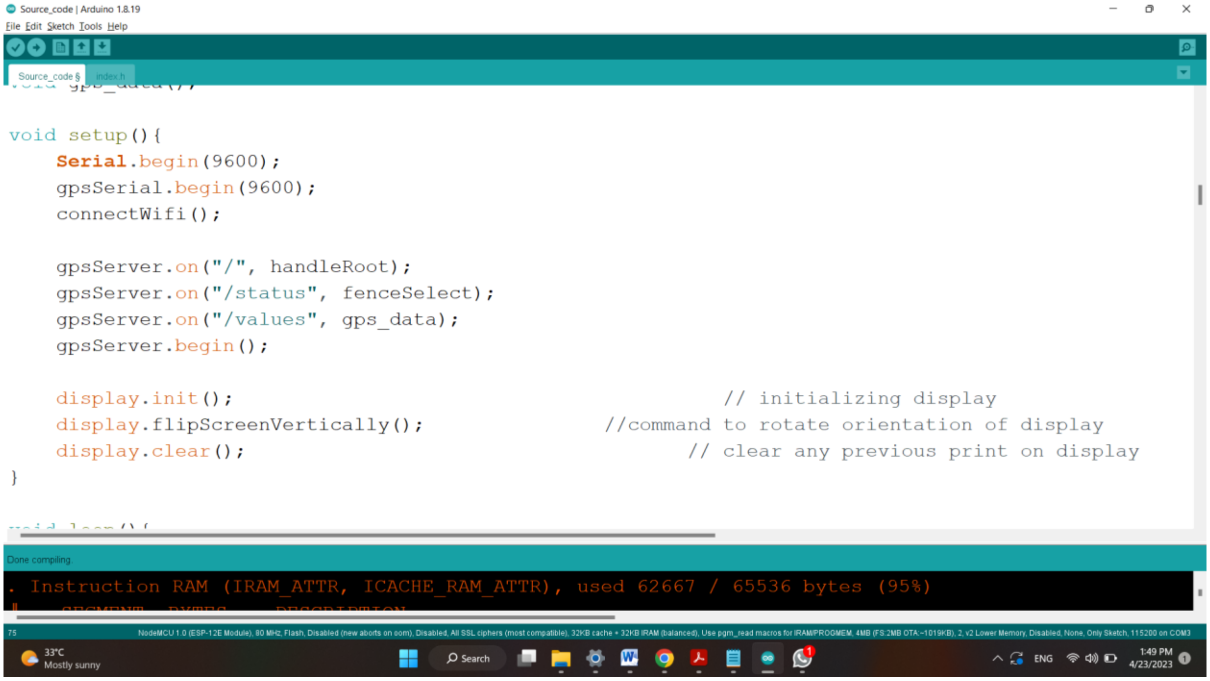
```
TinyGPSPlus gps;
ESP8266WebServer gpsServer(80);

void connectWifi();
void updateLatLon();
void pip();
void pip2();
void pip3();
void pip4();
void handleRoot();
void fenceSelect();
void gps_data();

void setup(){
    Serial.begin(9600);
```

Done compiling.

```
╟── DATA      1540      initialized variables
╙── RODATA    1916      constants
```

4 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

Calling the declared functions



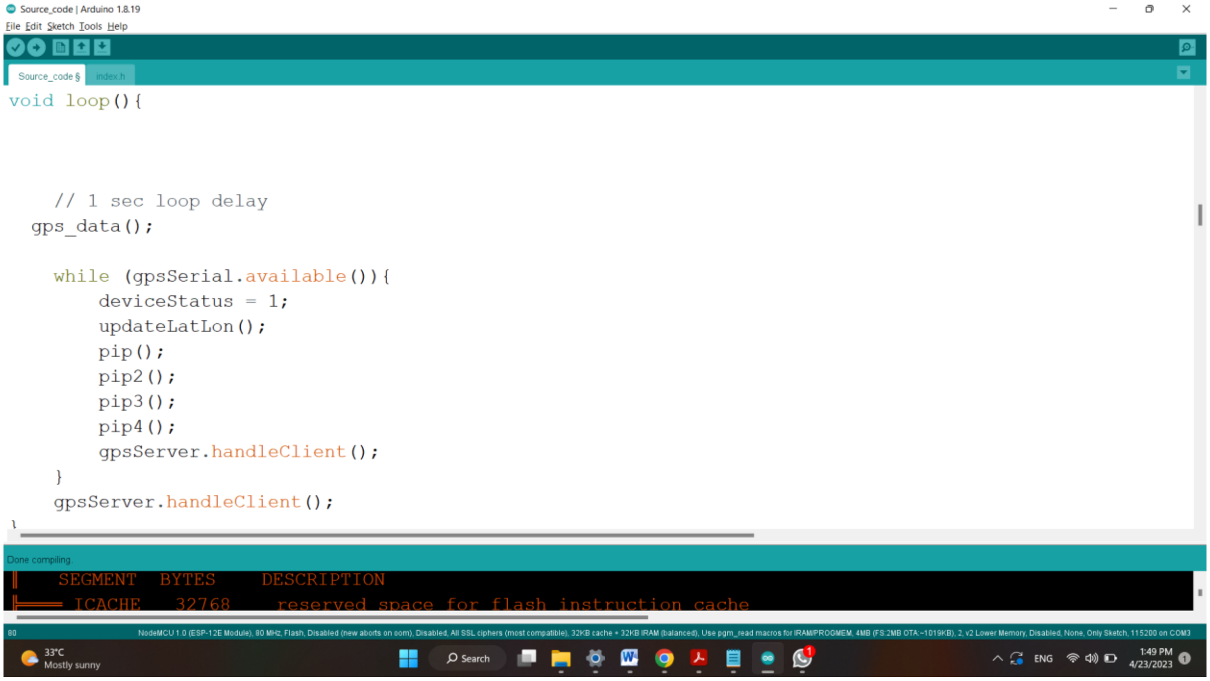
```
void gps_data();

void setup(){
    Serial.begin(9600);
    gpsSerial.begin(9600);
    connectWifi();

    gpsServer.on("/", handleRoot);
    gpsServer.on("/status", fenceSelect);
    gpsServer.on("/values", gps_data);
    gpsServer.begin();

    display.init();                                         // initializing display
    display.flipScreenVertically();                //command to rotate orientation of display
    display.clear();                                      // clear any previous print on display
}

void loop(){
```

Done compiling.

```
. Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 62667 / 65536 bytes (95%)
```

75 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

This code starts by setting up a serial connection with a baud rate of 9600 for communication with the other devices. The GPS Serial variable is also initialized with the same baud rate. The function then calls the connect WIFI () function/method to connect the device to a Wi-Fi network. The esp8266 WebServer library is used to set up the web server for the device, with different endpoints defined using the on () function. The function then initializes a display and flips the screen vertically to ensure proper orientation. Finally, By using clear() function we can clear the previous function.



```
void loop(){



    // 1 sec loop delay
  gps_data();

    while (gpsSerial.available()){
        deviceStatus = 1;
        updateLatLon();
        pip();
        pip2();
        pip3();
        pip4();
        gpsServer.handleClient();
    }
    gpsServer.handleClient();
}
```

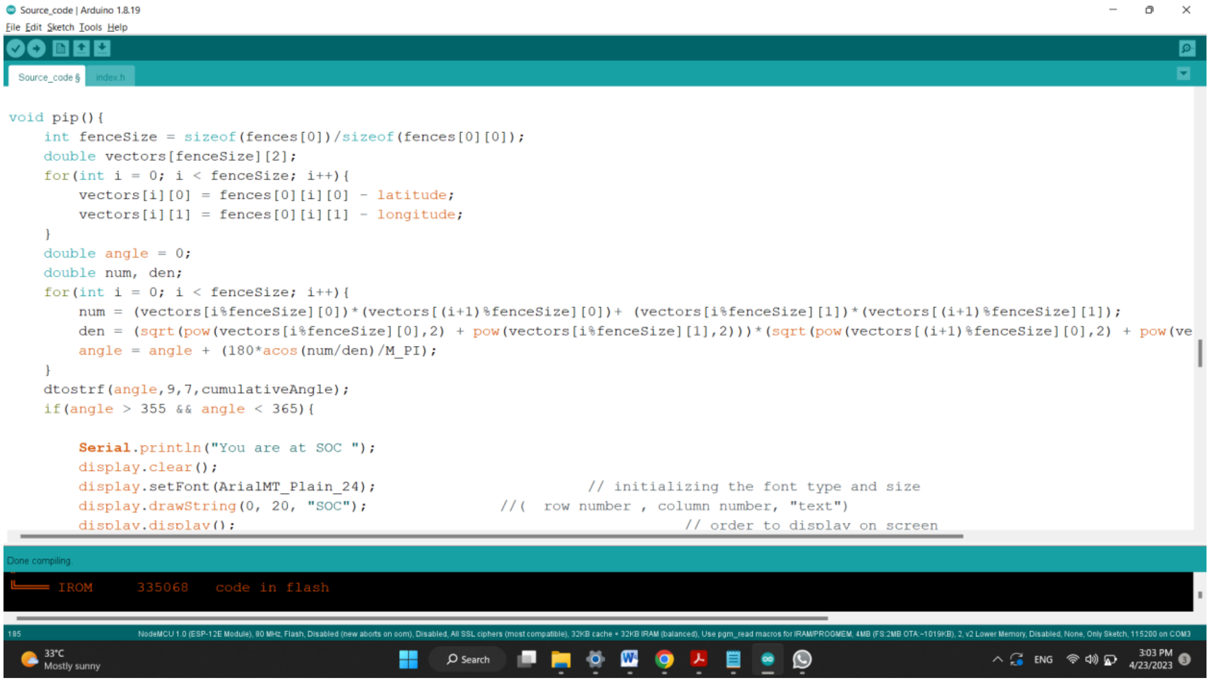
Done compiling.

```
║    SEGMENT  BYTES     DESCRIPTION
╠═══ ICACHE    32768     reserved space for flash instruction cache
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

Until the GPS signal is true the pip functions will be in loop .

Pip function is the function which will give the user to identify the user's current location and it will display the user's current location name in the iot product.



```
void pip(){
    int fenceSize = sizeof(fences[0])/sizeof(fences[0][0]);
    double vectors[fenceSize][2];
    for(int i = 0; i < fenceSize; i++){
        vectors[i][0] = fences[0][i][0] - latitude;
        vectors[i][1] = fences[0][i][1] - longitude;
    }
    double angle = 0;
    double num, den;
    for(int i = 0; i < fenceSize; i++){
        num = (vectors[i%fenceSize][0])*(vectors[(i+1)%fenceSize][0])+ (vectors[i%fenceSize][1])*(vectors[(i+1)%fenceSize][1]);
        den = (sqrt(pow(vectors[i%fenceSize][0],2) + pow(vectors[i%fenceSize][1],2)))*(sqrt(pow(vectors[(i+1)%fenceSize][0],2) + pow(ve
        angle = angle + (180*acos(num/den)/M_PI);
    }
    dtostrf(angle,9,7,cumulativeAngle);
    if(angle > 355 && angle < 365){

        Serial.println("You are at SOC ");
        display.clear();
        display.setFont(ArialMT_Plain_24);                        // initializing the font type and size
        display.drawString(0, 20, "SOC");               //(  row number , column number, "text")
        display.display();                                                   // order to display on screen
```

Done compiling.

```
IROM      335068    code in flash
```

185   NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

This is a pip function in here we are dividing the stored locations latitude and longitudes in

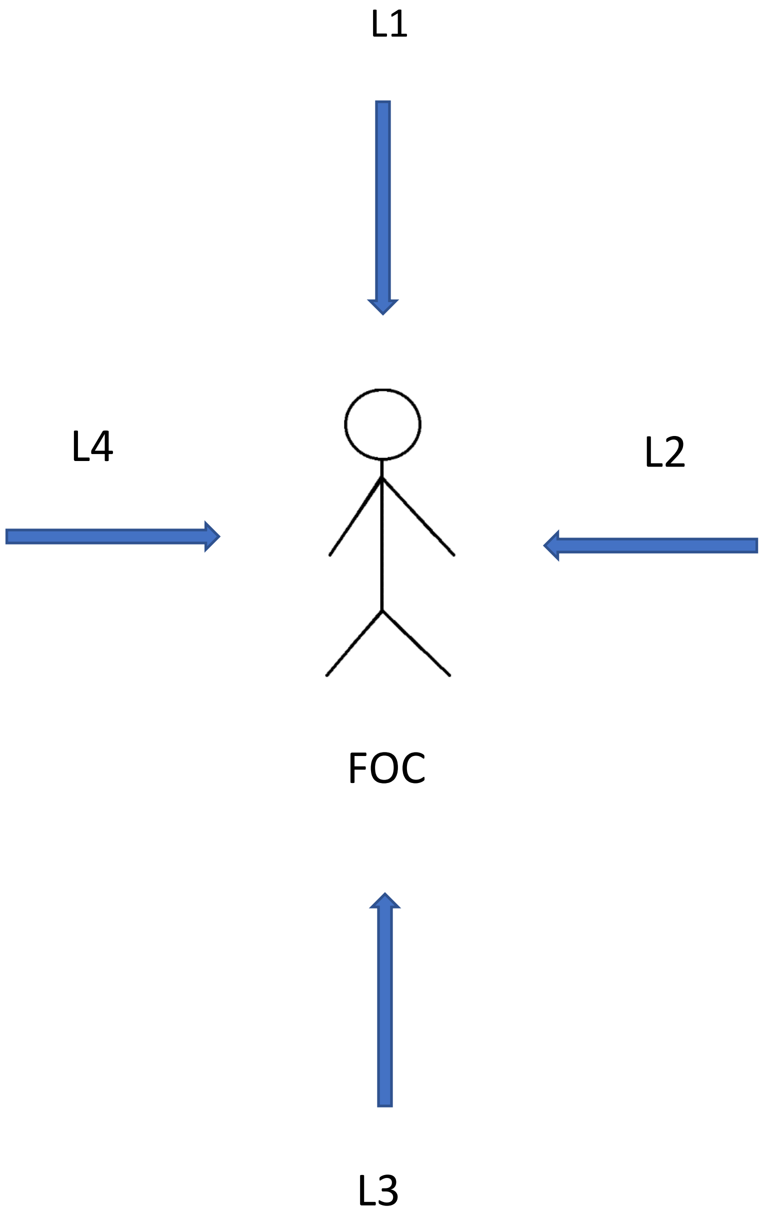
Vectors .and in here it is testing whether the user is in the right location (It's like double checking here) first we stored the right location latitude and longitude and then we are checking the user's location variable is getting the signal from the 4 defined corners(checking whether the signal is coming 360)

and if that condition is fulfilling we can display the user's location name

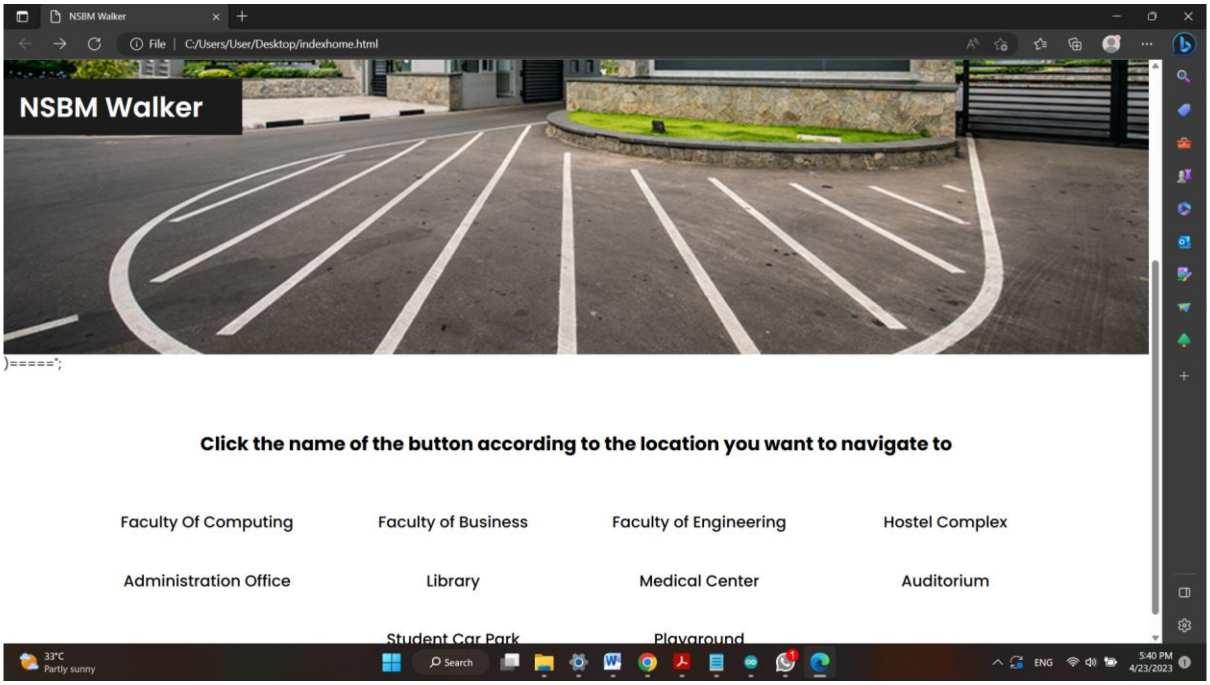
This code is for a GPS tracking system using an ESP8266 WiFi module, which receives GPS data via a GPS module connected to the RX and TX pins of the ESP8266. The GPS data is processed using the TinyGPS++ library. The code includes functions to connect to a WiFi access point, update the GPS data, and display it on an OLED display.

In addition to that when focusing on displaying GPS data, the code also includes a method/function to define a geo-fence around a specific location using latitude and longitude coordinates. This function will check whether the GPS data is inside the defined geo-fence and sets a flag accordingly. As descried in the hardware section This code also includes a web server to display GPS data and the status of the geo-fence flag. The web server will react on on port 80 and has three routes: root, status, and values. The root route displays a basic HTML page with links to the other two routes. The status route displays the status of the geo-fence flag. The values route displays the latitude, longitude, and number of satellites in view.

this is how it works



These are some screenshot of our web host

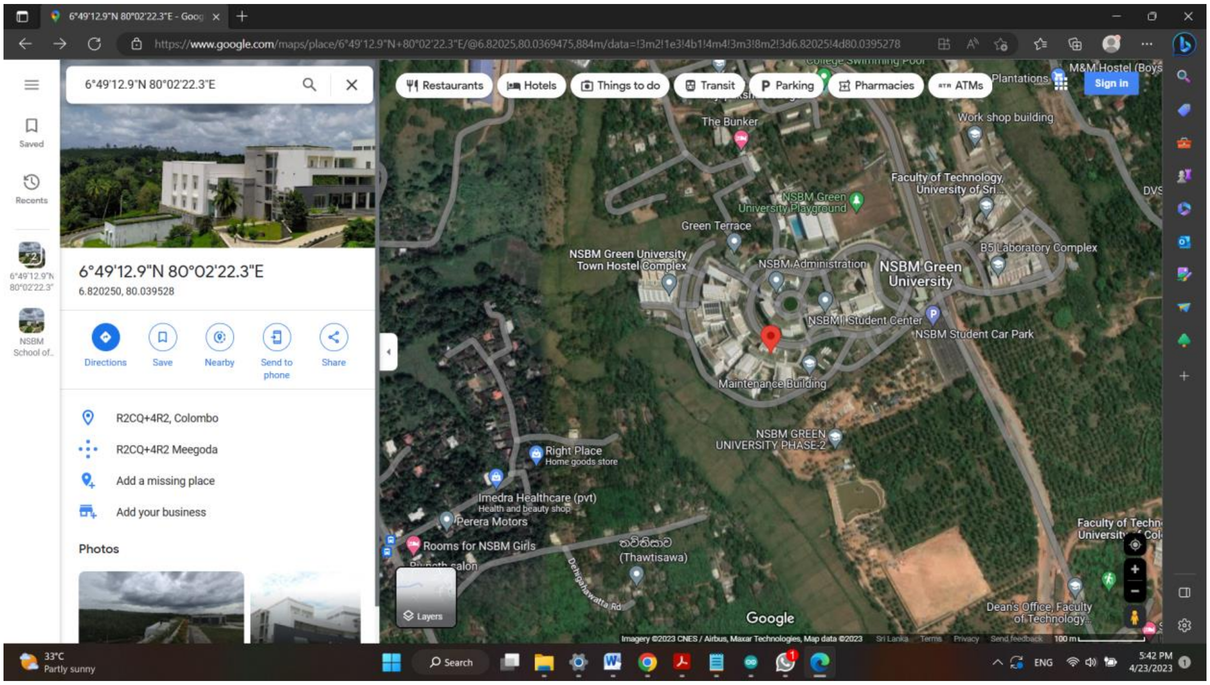


NSBM Walker

)=====";

Click the name of the button according to the location you want to navigate to

Faculty Of Computing
Faculty of Business
Faculty of Engineering
Hostel Complex
Administration Office
Library
Medical Center
Auditorium
Student Car Park
Playground



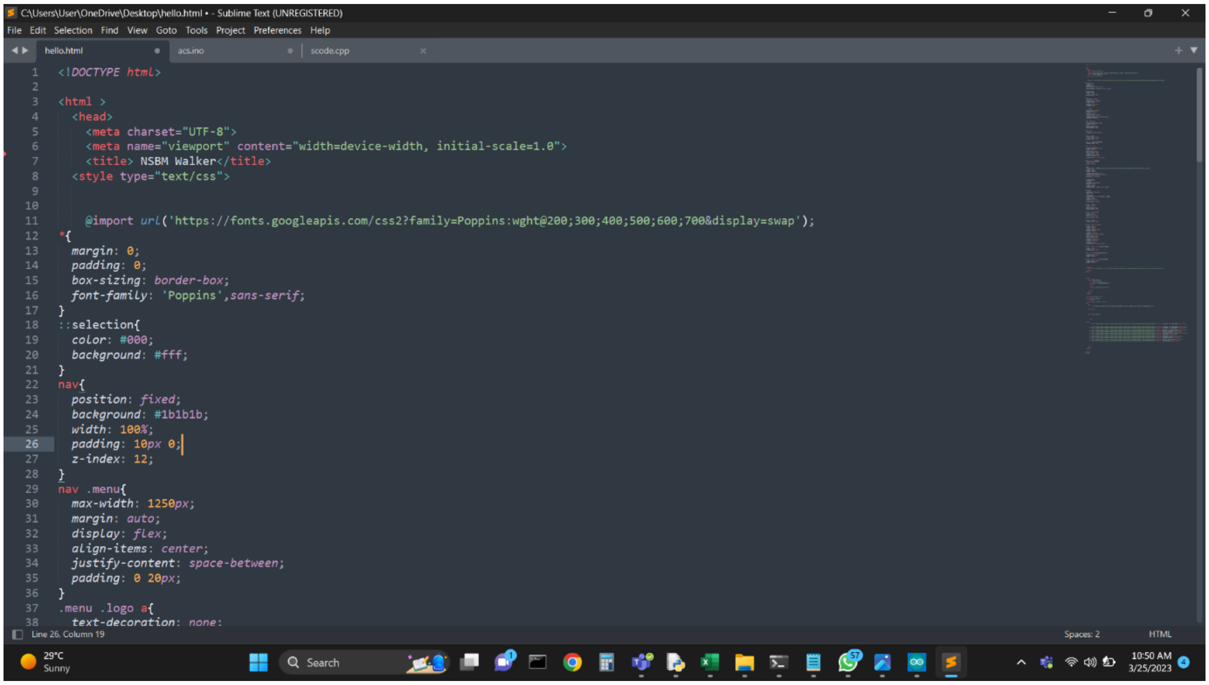
6°49'12.9"N 80°02'22.3"E

6°49'12.9"N 80°02'22.3"E
6.820250, 80.039528

Directions
Save
Nearby
Send to phone
Share

R2CQ+4R2, Colombo
R2CQ+4R2 Meegoda
Add a missing place
Add your business

Photos


C:\Users\User\OneDrive\Desktop\hello.html • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hello.html
acs.ino
scode.cpp
Line 26, Column 19
Spaces: 2
HTML
29°C
Sunny
Search
10:50 AM
3/25/2023

```
<!DOCTYPE html>

<html >
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> NSBM Walker</title>
  <style type="text/css">


    @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
::selection{
  color: #000;
  background: #fff;
}
nav{
  position: fixed;
  background: #1b1b1b;
  width: 100%;
  padding: 10px 0;
  z-index: 12;
}
nav .menu{
  max-width: 1250px;
  margin: auto;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
}
.menu .logo a{
  text-decoration: none:
```

For the web we used HTML, CSS, JavaScript. This is basically the front end development of the project.

This is the final iot product of our project with displaying the names

Workload Metris

Name	Plymouth Index	Contribution
Goniyamalaimage Aponsu	10820856	Code navigation software integrate GPS and display components, test software
Chaweeshwara jayasinghe	10820739	Research and select hardware components, design, and test circuitry, ensure hardware and power requirements
Gurusinghe A pankaja	10820854	Plan project schedule, coordinate team meetings and communication ensure deadlines are met
Thellambure Hettige Amantha	10820863	Design the visual layout and user experience for web page, create wireframes and mockups, gather user feedbacks
Kuda Arunasiri	10820737	Build web page using HTML, CSS AND JavaScript, integrate with navigation software, test web page
Rathnayake Rathnayake	10820945	Test the iot unit hardware and software, test web page functionality and usability, report defects.

Conclusion:

Our IoT device and through the QR code the user will direct to the web host.as the development team we believe that this have to be an effective ,innovative and efficient solution for the guidance and navigation problem that described above in the description . The University(NSBM campus), enabling visitors to navigate the premises with ease. By accurately tracking the user's location in real-time, the device provides a reliable and a user-friendly navigation experience, allowing users to quickly find their way around the university.

Looking ahead, we see significant potential for further improvements and enhancements to the device. For instance, As we educated earlier in the above description AR technology will take this project to the next level. It will create a more immersive and interactive experience for users, making it easier for them to navigate the campus and engage with its features And Not only that but also if AR technology is using in the system it will be much easier to display the indoor navigation too. Furthermore, developing a smaller and more portable device would increase its portability and convenience for users. Adding a touch screen which user can scroll up, down right and left will create more interesting display. It will also provide users with more detailed information about the university premises, allowing them to explore and engage with the premises in even greater depth.

These enhancements will definitely elevate the value of the device, particularly for guests visiting the university for the first time. By providing them with a more immersive and interactive experience, the device would enhance the overall quality of their visit and improve their understanding of the NSBM campus. In conclusion, our IoT device and QR code solution provide an excellent foundation for future innovations and improvements that will enhance the visitor experience and strengthen the university's reputation as a leader in innovative technology solutions.

References

- *M. McGrath, "Navigation unit with GPS technology, microcontrollers, and web hosting," GitHub repository, 2021. [Online]. Available: https://github.com/mmcgrath11/navigation-unit.

- *Arduino. (2021). Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/.

- Shivananda, P., Kamath, V. R., & Prabhu, G. K. (2018). A review of indoor positioning systems. International Journal of Engineering & Technology, 7(3.29), 611-614. https://www.sciencepubco.com/index.php/ijet/article/view/16441/7312

- Hong, W., Kang, B., & Yoo, H. (2019). Indoor navigation system based on an IoT-based sensor network for smart campus. Sensors, 19(20), 4511. https://www.mdpi.com/1424-8220/19/20/4511
- Khan, M. A., & Khan, W. Z. (2019). A smart campus navigation system based on the internet of things (IoT). Sensors, 19(21), 4658. https://www.mdpi.com/1424-8220/19/21/4658

- Toma-Danila, D., & Zeteanu, A. (2019, October). Improving campus navigation through a mobile augmented reality application. In 2019 IEEE 23rd International

- Api knowledge
- (https://developers.google.com/),

- "ESP8266 Datasheet," Espressif Systems, 2014.

- "Neo-6M GPS Module Datasheet," U-blox, 2016.

- W3Schools. (2021). HTML Tutorial. [Online]. Available: https://www.w3schools.com/html/.

- W3Schools. (2021). CSS Tutorial. [Online]. Available: https://www.w3schools.com/css/.

- Atlassian. (2021). Agile Methodology: The Complete Guide to Understanding Agile. [Online]. Available: https://www.atlassian.com/agile.

- Lucidchart. (2021). Use Case Diagram Maker. [Online]. Available: https://www.lucidchart.com/pages/use-case-diagram.