



Summer 2024
02-24-01205
Field Training 1
Internet of Things: Theory & Application

Prepared by:

{Ali Mohamed Ali Soliman}
{22011564}
{Internet of things (IOT)}

Started on:
september/1/2024

Delivered on:
september/11/2024

Supervised by:
Eng. Mohamed Hatem Abdulkader

Table of Content:

1. Introduction:
 - 1.1. C/C++ introduction
 - 1.2. Course Cycle and Timing.
2. C/C++:
 - 1.1. introduction to C's syntax and basics with the differences in it than any other language
 - 1.2. getting deep into C till right before OOP.
3. Esp 32:
 - 1.1. introduction to microcontrollers just like esp32 and differences what makes it a better deal.
 - 1.2. introduction to each hardware piece included in the kit and what they're used for.
 - 1.3. resistors and how to read them.
4. Cloud Computing with MQTT and Firebase:
 - 1.1 what's MQTT and difference between MQTT and HTTP, and what's better and why.
 - 1.2 introduction and briefing about HIVEMQ and how it's used and when.
5. Flutter application:
 - 1.1 introduction to dart.
 - 1.2 introduction to flutter.
 - 1.3 introduction to firebase.
 - 1.4 Flutter's UI and how to use it to create and deploy application
 - 1.5. MQTT with flutter and firebase
6. Bonus sessions:
 - 1.1. AI in IoT.
 - 1.2. IoT Security.

i. Training Methodology:

a. Interactive lectures:

- i. I've attended all offline at the university room 305 Saturday 9 o'clock to 1 and sometimes 2 or 2:30 AM, as for the online lectures I've attended most of them starting with the first one session zero which was the introduction to the course and a get to know all the course instructors what the course would be about and something of a discussion about what we should do and how many lectures would take us to finish a course and we agreed on the day and time of each of the 4 coming offline lectures and also the number and topics to be discussed in those lectures, as for the rest of the online lectures I've attended most of them which were on Thursday mostly 7PM to 9 and sometimes 10PM which were more of a practice and a query session in which we would ask any of the questions we wanted to know and the rest of the session was a volume 2 of the offline session we took the same week with topics that relate to Saturday's topics and more genuinely a practice on that we learned on the previous offline session, and for the rest of the online sessions were mostly recorded so if I didn't have the time to attend the live sessions I'd go and watch the recorded session any day before the next Saturday so I'd be on track with what's going on and what we've learned in those sessions.
- ii. **& III.** Well as I said session zero: was an introduction and we talked about c/c ++ which is the coding language we'll be using throughout our course to develop and write down the code which would be uploaded to the esp32 to control the hardware as we wanted to... session 1: the first offline session we attended and was all about the c/c ++ syntax, style, what's the difference between it and every other coding language we took before just like java and python and what we've learned was that it's the hardest language of all of them and that's reasonable because it's the oldest among all of them and one of the oldest languages of all time actually as all what we know and use now is most likely built with c as it's one of the closest language to computer language (binary) and has the fastest compiler of most of the other languages well except for assembly which c is compiled down to it as it's the first and hardest language ever made, we also learned about pointers which was a new topic to hear about compared to java and python which also contain pointers but it's so much different in c it practically allows you to create a pointers that points directly to the place saved in memory for your variable which is called an address and we had some practice 3 questions to be exact that I solved containing 2 questions in which I used pointers and the other one was a simple question about the fundamentals of c or any programming language to be honest in which I used stars to build a pyramid using a simple nested for loop and then the updated version was to use numbers

sequentially to build the pyramid which was easy considering that we're using a for loop with an already defined counter.... 1 online session: which as I said before was a continue session and a discussion were we practiced so many questions about c and everything that's new or different about it and we had some answers to some of our questions that helped pointing us in the right direction and we stopped right before we got into the OOP of c which we didn't need in our course... session 2: which was the second offline session we attended and was all about the hardware and the esp32 which we'll be using throughout our course and we had already bought all the hardware we needed to use in the course as we've been instructed to do in a delivery message containing all the hardware pieces we'll be using and for that sessions the first half was theoretical that would get us to know what is the esp32 and its advantages compared to other chips like the most known Arduino uno 3 which was at the same price or maybe even more expensive and doesn't have the WIFI or Bluetooth and so the esp32 was a much better deal especially for beginners who had no technical or electrical background as it was easy to deal with and we also learned a lot about the rest of the hardware pieces included just like the resistors and how to read them with the lines drawn on each of them so you'd know which to be used when needed and we also had some assignments that would be handed over at the end of the session including what we've learned and it was as hard as it gets at the time that we had to connect the esp32 to the servo motor and a keypad in which every button would get the servo to move to a specific angle and to be honest that's my favourite session .. 2 online session: was also more of a query in which most of our questions were answered and we had some practice of the hardware connections and how to use platformio and wokwi to create an online simulator of our circuits which would make it much easier when we get to physically connect those hardware... session3: which was one of the hardest and most difficult sessions that we ever had in which we learned about the global configuration of the esp32 and how to use a broker just like hivemq or mosqueutto to upload our code to it directly and then using the callback function would get those readings to be printed on the serial monitor or even be used to do things like moving a servo or make the buzzer make a sound or not and I had a lot of errors that day lots and lots of problems in which I didn't know how to handle or where they were coming from especially when I tried to use the private broker instead of a public broker and those errors were happening to all of us and weren't solved until the next online session... 3 online session: it was the session that we were all waiting for to answer our questions and to tell us how to deal with all those errors and that's exactly what happened we had some theoretical discussions and then we got to the hardware and wokwi where most of our questions were answered and errors were solved just like to. Use a private broker we had to use the library client secure and to add a certificate to our code that would allow the

esp32 to access the private broker I created and in this session we also learned some thing about the final project and that we need to take the global esp32 session seriously because we'll most definitely use it in it and writing about it know I didn't just use it's more like 50% of the final project... 4 (last) offline session: before this sessions there were 2 more online sessions to explain dart and talk about the final project requirements which I attended on time and the really helped to explain and clear out some of the things I had in mind for the project and the flutter application and it's use in all of that, getting to the offline session itself we learned some of the things we needed to know about flutter just like stateful and stateless widgets the decorations and the UI of flutter just like containers, columns, raw ...etc and buttons and of course we had assignments to be handed over by the end of the session 2 to be exact 1 to get us to know more about the containers and the decorations and another to teach us about text fields and how to deal with them and all that remains was the firebase and it's connection to flutter and how to use it in the final project for authentication.. the final (4) online session: it was a session completely including everything we've learned and took in all of the previous sessions including the errors we've encountered in the esp32 hardware and with the esp32 global getting to flutter and the UI and finally the firebase and how to connect it to flutter for authentication and even more that if we wanted to use it to send and receive the data instead of the MQTT we could .. there were 2 more online and offline sessions about

b. Learning by doing:

- i. for session zero there were none ... session1 : there were 3 questions to be handed over by the end of the session one of them as I've mentioned previously was to print a pyramid of stars using a nested for loop and then the updated version of it was to print the pyramid of numbers sequentially starting from 1 using the same for loop which was easy considering that for loop has an already defined counter of integers so I used it to print the pyramid and the other 2 were about using pointers to change the value of a certain variable without creating a new one ... session 2 : it also included questions to be handed over by the end one of them was to light a led using esp32 jumper wires and a 220 ohm resistor and another was to connect a servo motor to the esp32 with a keypad and configure some of the keys on the keypad to have a certain angle that when pressed the servo would turn to this specific angle and when the * key is pressed it goes back to zero... session3: as I mentioned before it was the most difficult session of all of them and so were the questions at the end of it as we connected the servo to the esp32 and connected the code the HIVEMQ broker and we were supposed to publish the degree that the servo is at at the moment and subscribe to the same topic to print the servo angle from the callback function directly to the serial monitor and when

another number is published manually the servo would turn to that specific angle ... session4: it was a light session with 2 questions to be handed by the end of it one was to print out all the names of my team into the screen of flutter app and wrap them with a container and change the container's background colour and the other one was to create 2 elevated buttons in the middle of the screen and when any of them is pressed it navigates us to another page with a text field input just like a publish and subscribe to MQTT.

- ii. **&III.** I learned about pointers in the first session which was a new topic to deal with especially in c as it points directly to the address of the variable and also arrays as they're in some way different in c to deal with than any other language, I also learned about connecting the resistors in series and in parallel by the second session and how to read the resistors by the lines that're drawn on each of them and how to connect each of the hardware pieces to the esp32 where does the VCC connect to where does the GND connect to and the types of pins on the esp32 which every hardware should be connected and by practice I learned which hardware should be connected digital and which analog and which is to be connected to the ADC (analog to digital converter) pins, by session 3 I learned about the MQTT the brokers what they're used for and when how to publish subscribe to them how to create a cluster with a unique credentials to be my private broker and use it to publish the readings to it and then subscribe to the topics which the data has been published to, I also learned how to connect the esp32 to the WIFI and that I cannot use the ADC2 pins when the esp32 is connected to the WIFI because it uses it, by session 4 I learned about flutter the UI, decorations stateful and stateless widgets and when they're used I also learned about buttons and navigation, by the last online session I learned all about the firebase and how to connect it to flutter to use for authentication or even to replace the MQTT to send and receive data using real time database and how to connect both flutter to the fire base and esp32 to the flutter app to send receive and view sensor readings in real time as that's what we'd use in our final project.

c. Continuous assessment and peer learning:

- i. I've worked on all the assignments we've had and did most of them and consulted with my team mated and peers as for assignment 1: I had already a background of c and c ++ and so it wasn't so hard to achieve and solve all the questions included in that assignment but before I did that I had to remind myself of c first and research pointers to know more about it and how to deal with them and as a team leader I told all of my team mates to do that also next to learning the syntax of c and that we'd give it 2 days for research and study and then we'd start to divide the assignment questions equally on all of us so we'd all learn it and that would help us throughout the rest of the course and 2 or 3 days

passed and then I divided all the questions equally upon us and I decided that all of us should research and do the theoretical question so we'd all learn and know more about the esp32 and how to use it and by that time I had done all the questions because of my background of c and waited for all my team mates to be done with their parts and that if any of them had a question I'd help with it and by the deadline we were done with it all with the bonus questions and submitted it on the classroom... assignment 2: it was a very big assignment with many questions of c , theoretical and esp32 hardware implemented questions and so we've landed on that each of us should take a topic of those to deal with so some of the team had the theoretical questions the others had the c code questions and I had the esp32 with the hardware questions and we started working on it and every one would finish would write down a report of what he did and I did the same and add the videos of each of the questions with a schematic for each circuit and a wokwi session of the same hardware implemented code that I started with to make everything easier and that took me quit q while and so I told all of my team to go to wokwi and try to implement the code on the simulator to give them a better understanding of the connections , hardware and applying the code and uploading it to the esp32 ... assignment 3: there were some theoretical questions and so my team decided to handle that part while I handle the esp32 questions along with the esp32 global questions and that's what I did I had to research a lot of things including MQTT , HIVEMQ the configurations of the code and how to use the private broker instead of the public broker and I did use the public broker of mosqetto in some of the questions because it was much better than HIVEMQ public broker and I researched the node red and how to use it along with the MQTT and I also used the MQTT explorer to view the readings and I made videos of every questions of those along with a schematic of every one of them included in the report along with the wokwi sessions of each question ... assignment4: my team were studying for the final project and so I had to deal with the flutter application assignments and they weren't so hard but so many so did most of them along with the theoretical questions and we consulted on some of them .

- ii. **&III.** I've learned a lot of thing working all through those assignments most of all was that the hardware connections is the simplest part of any problem you have to go through a lot of stages till you get to physically implement your idea on the real hardware for example you have to start by researching all the hardware you'd need and how it works which of them should be connected digitally and which should be analog and why and when you're done you have to start to write down the code for each piece of hardware on it's own to check if it's working properly and to try a lot of test cases to know for sure that no errors would hit you in the face after adding it to the complete code and to repeat that with every other piece of hardware and when you're done you have to apply all that

on a simulator so nothing would burn out or be damaged of your real hardware and when all of that is done you get to apply all that to your real hardware and check for test cases... as for the topics I've learned there's quite a lot actually pointers and arrays in c and c ++ , setup and loop functions with the esp32 , the hardware connections , types of pins on the esp32, global esp32 with the MQTT , difference between HTTP and MQTT , HIVEMQ clusters and private broker , HIVEMQ certificate to access the private broker , node red and what's it's used for and how to use it with the HIVEMQ , WIFI with the esp32, dart and flutter with the widgets , buttons , decorations...etc.

d. **Capstone project:**

- i. the idea of our project was to implement and apply all what we've learned throughout the course to create a smart home with sensors implemented all over the home itself that have reading and those reading are viewed and monitored with a flutter application of my creation where you also get to control all the motors implemented all over the house to open and close the doors and also control all the LEDs ... as a view of what we wanted it to be is : a villa surrounded with a fence and an outside door that opens using the RFID sensor and when opened there would be lights on the ground leading to the main door that would light up using the motion sensor (PIR) until you get to the main door and there you have 3 trails to enter a code on the keypad and if its right the main door will open up and the LCD will print a greeting message just like : hello or welcome and if the code is wrong for all of the 3 trails a buzzer would make a sound and LCD would print a warning message just like : intruder!!! And when you get inside the house the outside lights would turn off also would find flame sensor that when a flame is detected would turn on the buzzer and the reading would show on the flutter app in real time same for a gas (MQ4) that is connected as analog that sends a warning message and turns on the buzzer if the gas levels exceed a certain threshold that's already been set and a rain sensor that shows if there's rain or not .. with the flutter app specified for this house you get to control all the doors, windows and LEDs in the house ... also I didn't just want to make anything I wanted to create everything from the house itself which I worked on very hard to make it look nice not just a box but a home that would really look like a maquette that could be applied to real houses and not made of carton but wood to give us a more premium and real looking house and that took me three days to made with a help of a carpenter and designer to help be implement all the ideas I wanted to and then I got home to implement all the motors on each of the doors and the window with wooden made hinges for each of the them so it would be easy for each motor to control the doors also to find the best places to implement each of the sensors so the readings wouldn't affect each other.

- ii. I went to the carpenter to make the design and apply all the ideas and extras that I wanted to achieve and after creating the house itself with the fence doors and everything my team mates and I went in to implement all of the sensors and motors and motors as discussed and then we started working on the code to control all these sensors and motors that would be uploaded to the Esp and as I've said in the videos we had a lot of problems in this part as 3 ESPs were burnt out with no idea why or what happened everything was perfectly compatible with the others and I've worked on them before but there was a problem so we started taking down some of the sensors one by one to see where the problem is coming from still didn't work out until I figured out the issue and will discuss it in the project chapter but we had to remove some of the sensors to make it work with the 4th Esp
- iii. I have achieved most of what I wanted to achieve and designed but we removed some of the sensors we wanted to use just like the RFID, LDR, IR, KEYPAD, LCD because we only had one Esp left and we wanted to make the most of it so we connected 3 servo motors 1 for the main door , 1 for the garage door and 1 for the window along with 3 LEDs one in each of the rooms Living , Garage and kitchen and 4 sensors MOTION (PIR), RAIN, GAS (MQ4), FLAME all over the house and each of those are controlled monitored via the flutter app I've created

ii. Training content:

- 1. chapter 1: Introduction
 - b. the aim of the training was to learn each of the topics provided throughout each of the chapters and to be able to apply it on real life hardware and project
 - c. we used PLATFORMIO and ARDUINO framework to apply and upload the code to the Esp and we also used WOKWI as an online simulator and fritzing for the schematics.
 - d. The training cycle took a moth and two weeks for bonus and extra sessions: Session 0: introduction and a getting to know the instructors and what we'll be doing throughout the training along with an introduction or rather a brief of the to be used coding language, frame work and the Esp. Session 1: introduction to c and c ++ with its syntax and everything from loops, functions, conditions and pointers followed by an online session: to complete the introduction of the c and solve questions about c and more about pointers and the difference between c and any other coding language Session 2: hardware session about the Esp and each of the hardware pieces we'll be using in the next few

session also followed by an online session : to talk more and more about the Esp and solve questions about it on WOKWI and answer some of our questions Session 3: global Esp session that talked about the broker and how to use it along with the Esp and MQTT and the differences between HTTP and MQTT .. etc also followed by an online session: to help us understand the MQTT and how and why we use it with the HIVEMQ with a private broker Session 4: there was 2 recorded sessions before it about dart, flutter and the final project. And it was a continuance of those sessions to talk about flutter ad UI followed by 2 extra sessions to talk about security and AI with 2 online sessions to follow up on them.

- e. Well by all means it was hard a new topic to learn which was far from all we've learned in the previous 2 years took me quite a while to understand it and be able to somehow improve and evolve as the sessions go by and know from where I'm sitting I can say it's worth all that because as difficult as it was I really enjoyed being a part of it learning it and getting to see with my own to eyes what've done and have been working on move or work exactly as I wanted to and even better it wasn't just about writing a code and being done it was about making something from the start to the end and seeing how it works evolves and it was really truly amazing and I found myself and my passion in it .
- f. Session 0: introduction and a getting to know the instructors and what we'll be doing throughout the training along with an introduction or rather a brief of the to be used coding language, frame work and the Esp. Session 1: introduction to c and c ++ with it's syntax and everything from loops, functions, conditions and pointers followed by an online session: to complete the introduction of the c and solve questions about c and more about pointers and the difference between c and any other coding language Session 2: hardware session about the Esp and each of the hardware pieces we'll be using in the next few session also followed by an online session : to talk more and more about the Esp and solve questions about it on WOKWI and answer some of our questions Session 3: global Esp session that talked about the broker and how to use it along with the Esp and MQTT and the differences between HTTP and MQTT .. etc also followed by an online session: to help us understand the MQTT and how and why we use it with the HIVEMQ with a private broker Session 4: there was 2 recorded sessions before it about dart, flutter and the final project. And it was a continuance of those sessions to talk about flutter ad UI followed by 2 extra sessions to talk about security and AI with 2 online sessions to follow up on them.
- g. he idea of our project was to implement and apply all what we've learned throughout the course to create a smart home with sensors implemented all over the home itself that have reading and those reading are viewed and monitored with a flutter application of my

creation where you also get to control all the motors implemented all over the house to open and close the doors and also control all the LEDs ... as a view of what we wanted it to be is : a villa surrounded with a fence and an outside door that opens using the RFID sensor and when opened there would be lights on the ground leading to the main door that would light up using the motion sensor (PIR) until you get to the main door and there you have 3 trails to enter a code on the keypad and if its right the main door will open up and the LCD will print a greeting message just like : hello or welcome and if the code is wrong for all of the 3 trails a buzzer would make a sound and LCD would print a warning message just like : intruder!!! And when you get inside the house the outside lights would turn off also would find flame sensor that when a flame is detected would turn on the buzzer and the reading would show on the flutter app in real time same for a gas (MQ4) that is connected as analog that sends a warning message and turns on the buzzer if the gas levels exceed a certain threshold that's already been set and a rain sensor that shows if there's rain or not .. with the flutter app specified for this house you get to control all the doors, windows and LEDs in the house ... also I didn't just want to make anything I wanted to create everything from the house itself which I worked on very hard to make it look nice not just a box but a home that would really look like a maquette that could be applied to real houses and not made of carton but wood to give us a more premium and real looking house and that took me three days to made with a help of a carpenter and designer to help be implement all the ideas I wanted to and then I got home to implement all the motors on each of the doors and the window with wooden made hinges for each of the them so it would be easy for each motor to control the doors also to find the best places to implement each of the sensors so the readings wouldn't affect each other.

- h. I got to see how everything around as that's smart works and by the end of all those sessions now when I look at something that has IOT build system applied to it I feel and know how I could do it myself what it needs how much times and even some of the errors and issues that I'm going to encounter just simply by looking and it feels amazing that wasn't a brief but it is a sum up of all that I've learned .

2. Chapter 2: Training topics & practices

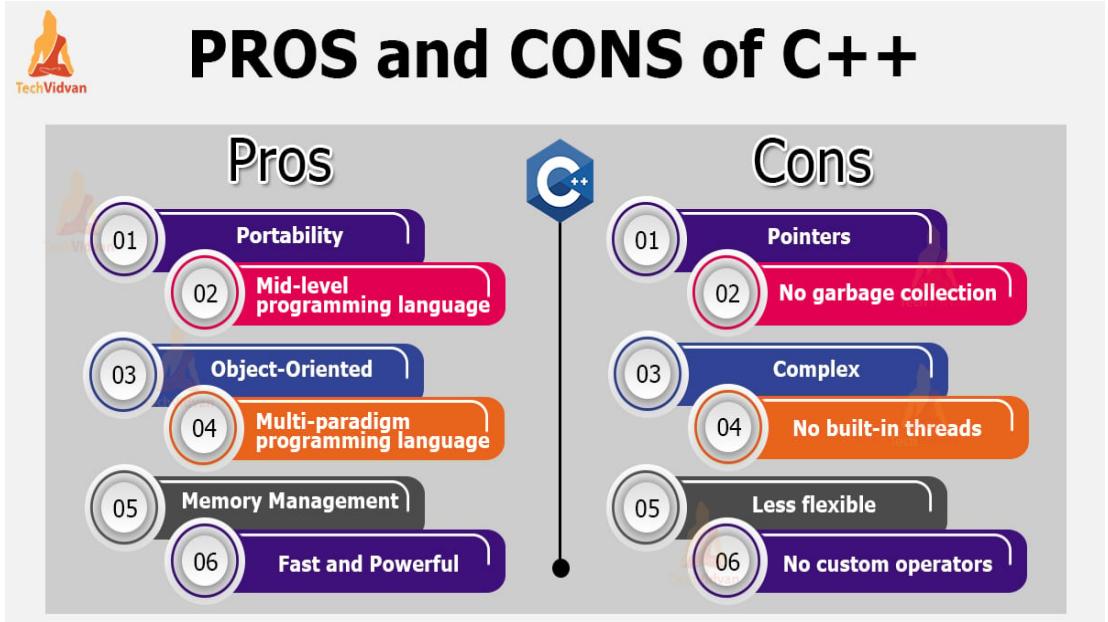
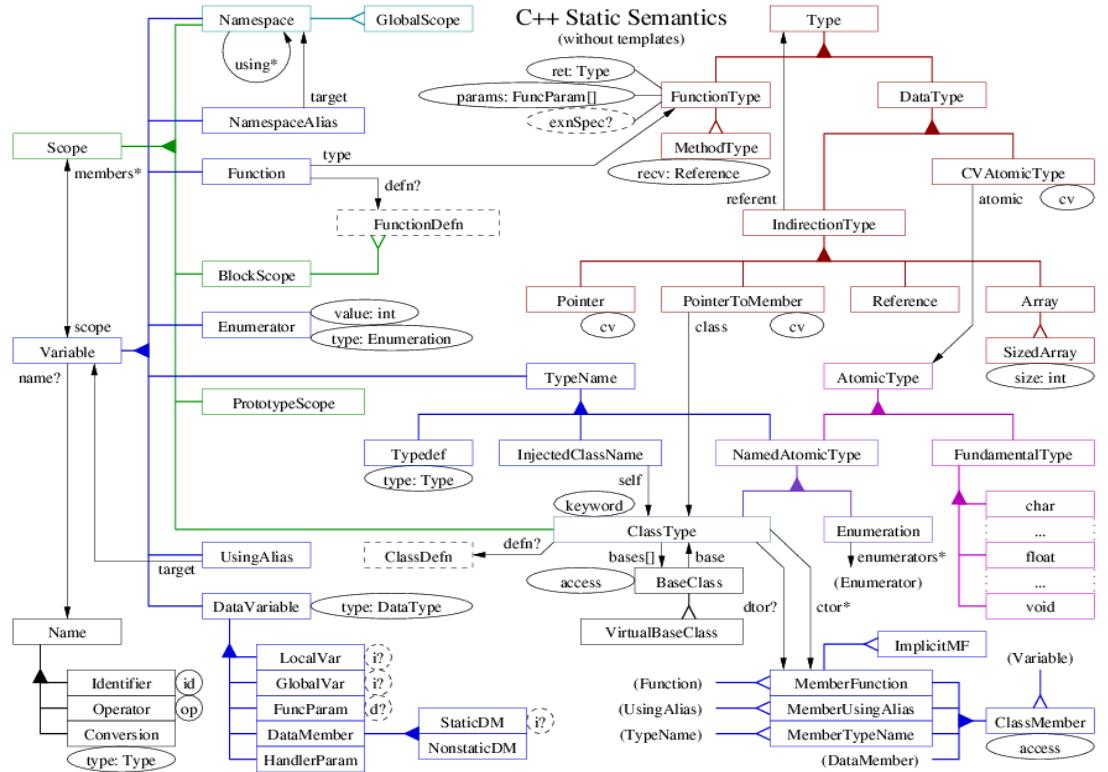
a. C/C++ Programming:

- i. I already had a background of C/C++ so it didn't take that much time so probably 10 hours.
- ii. Dr/ Mohammed Hatem.
- iii. I learned a lot about C's syntax and more about pointers and how to deal with them in C and the basic principles of

every programming language functions, loops, conditions...etc.

iv.

v. EX:

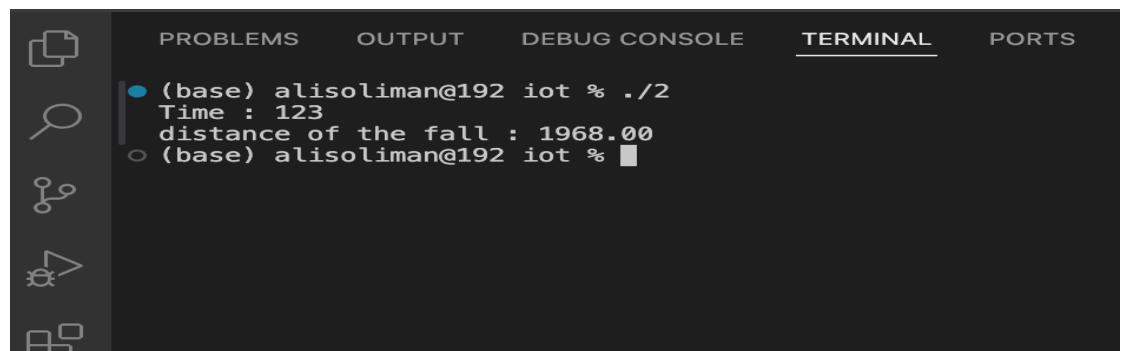


vi. I've worked on all the questions regarding C either on site where we had to deliver the 3 assigned questions by the end of the first session or the first assignment we had on site : the pyramid with the * and the updated version of it with the numbers the other 2 questions where we had to use pointers to interchange 2 variables without creating a new

one... as for the online questions I had the time so I worked on every last one of the questions included in the assignment.

- vii. The same questions are in the assignment so I'll include them below
- viii. I've worked on all the assignment question per my background and because I had the time.
- ix. The 2nd question in the assignment:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Time : ");
6     float time, distance;
7     scanf("%f", &time);
8     distance = (32 * time) / 2
9     ;
10    printf(
11        "distance of the fall : %.2f
12        \n"
13        ,distance);
14 }
```



The 3.1 question in the assignment:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("starting temp : ");
6     int temp
7     scanf("%i", &temp1);
8     printf("end temp : ");
9     int temp
10    scanf("%i", &temp2);
11    for int i = 0; i <= temp2; i++)
12    {
13        float velocity;
14        velocity = 331.3 + 0.61 * i;
15        printf(
16            "At %i degree celsius velocity of sound is %.1f m/s \n"
17            , i, velocity);
18    }
19
20 }
```

The output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● (base) alisoliman@192 iot % ./3
starting temp : 123
end temp : 12
At 0 degree celsius velocity of sound is 331.3 m/s
At 1 degree celsius velocity of sound is 331.9 m/s
At 2 degree celsius velocity of sound is 332.5 m/s
At 3 degree celsius velocity of sound is 333.1 m/s
At 4 degree celsius velocity of sound is 333.7 m/s
At 5 degree celsius velocity of sound is 334.4 m/s
At 6 degree celsius velocity of sound is 335.0 m/s
At 7 degree celsius velocity of sound is 335.6 m/s
At 8 degree celsius velocity of sound is 336.2 m/s
At 9 degree celsius velocity of sound is 336.8 m/s
At 10 degree celsius velocity of sound is 337.4 m/s
At 11 degree celsius velocity of sound is 338.0 m/s
At 12 degree celsius velocity of sound is 338.6 m/s
○ (base) alisoliman@192 iot %
```

The 3.2 question :

```
1  #include<stdio.h>
2  // prototypes
3  void curve(char letter);
4  void middle(char letter);
5  void line(char letter);
6  void mid(char letter);
7
8  int main(void)
9  {
10 {   printf("letter : ");
11     char letter ;
12     scanf("%c", &letter);
13     printf("yours : \n");
14     curve(letter);
15     middle(letter);
16     line(letter);
17     middle(letter);
18     curve(letter);
19     printf("\nbetter : \n");
20     curve(letter);
21     mid(letter);
22     line(letter);
23     mid(letter);
24     curve(letter);
25
26 }void curve(char letter)
27 {
28 {   printf("  ");
29     for(int i = 0; i < 3; i++)
30     {
31         printf("%c ",letter);
32     }
33     printf("\n");
34
35 }void middle(char letter)
36
37 {   printf("  ");
38     printf("%c      %c\n",letter,letter);
39
40 }void line(char letter)
41
42 {   for(int i = 0; i < 4; i++)
43     {
44         printf("%c\n",letter);
45     }
46
47 }void mid(char letter)
48
49 {   printf("  ");
50     printf("%c\t%c\n",letter,letter);
51 }
```

The output:

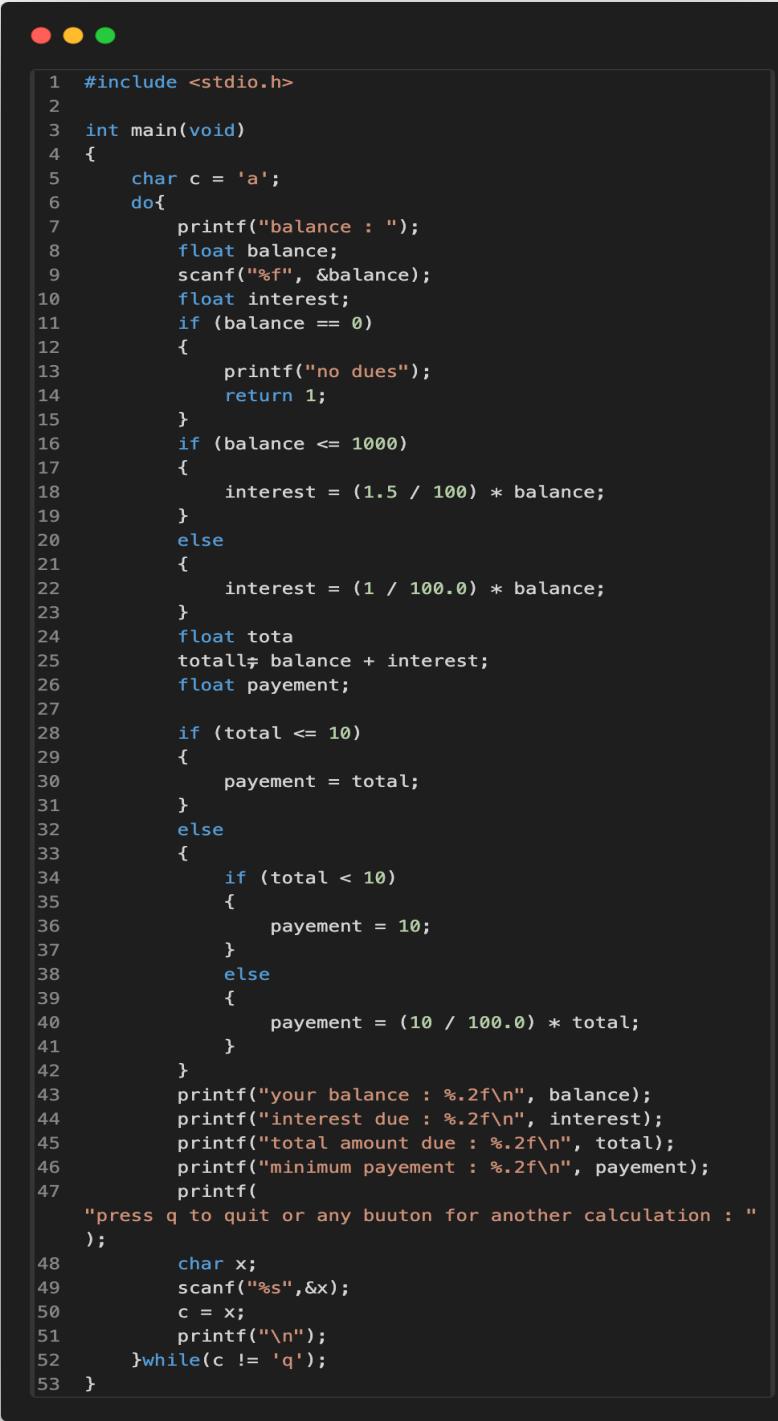
A screenshot of a terminal window in Visual Studio Code. The window has a dark theme and shows the following text output:

```
● (base) alisoliman@192 iot % ./c
letter : x
yours :
    X X X
    X      X
    X
    X
    X
    X
    X      X
    X X X

better :
    X X X
    X      X
    X
    X
    X
    X
    X      X
    X X X
○ (base) alisoliman@192 iot %
```

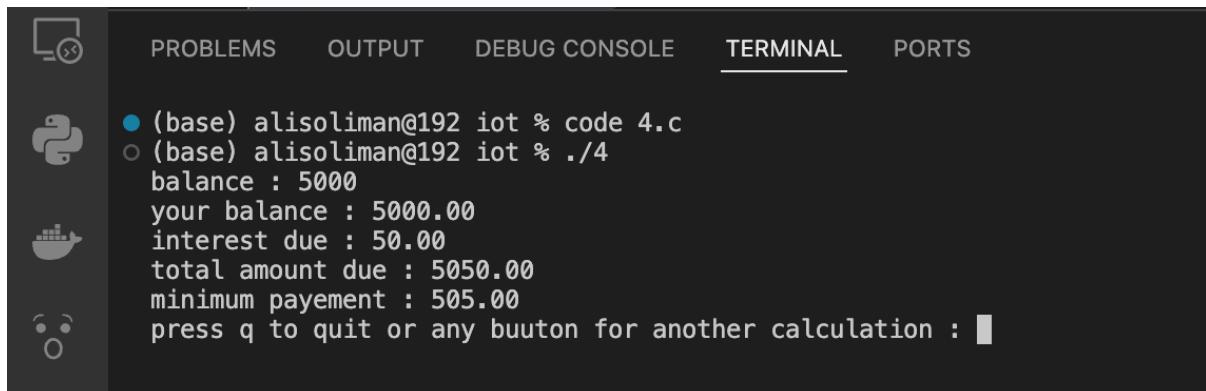
The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. There are also small icons on the left side of the terminal area.

The 4th question :

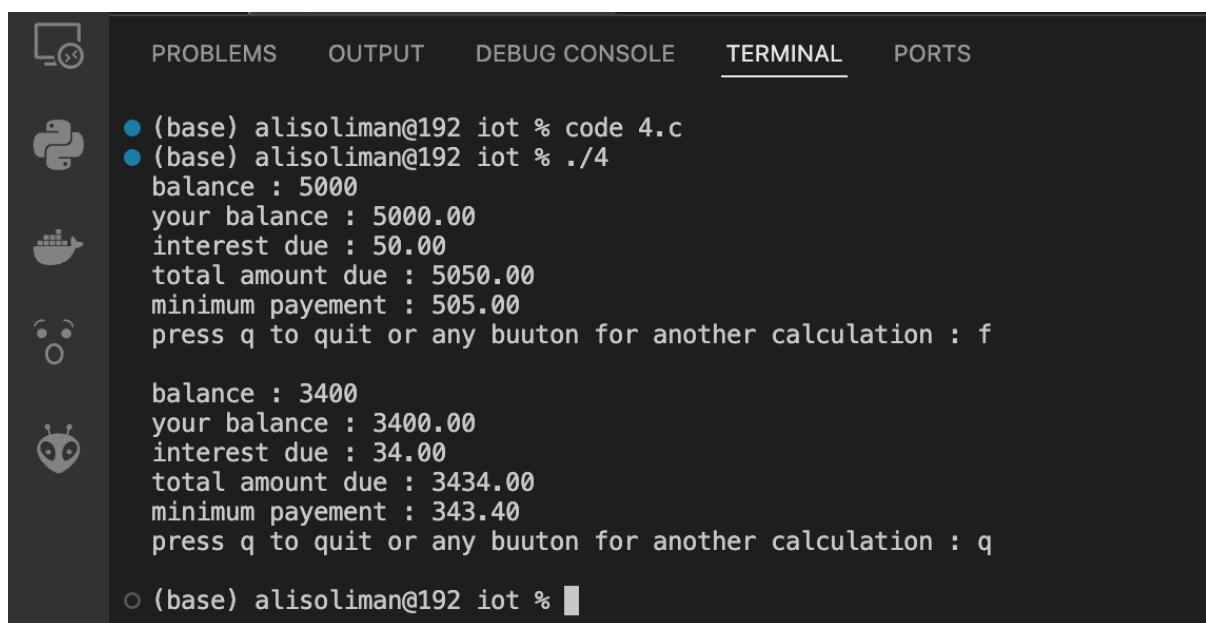


```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     char c = 'a';
6     do{
7         printf("balance : ");
8         float balance;
9         scanf("%f", &balance);
10        float interest;
11        if (balance == 0)
12        {
13            printf("no dues");
14            return 1;
15        }
16        if (balance <= 1000)
17        {
18            interest = (1.5 / 100) * balance;
19        }
20        else
21        {
22            interest = (1 / 100.0) * balance;
23        }
24        float total
25        total = balance + interest;
26        float payment;
27
28        if (total <= 10)
29        {
30            payment = total;
31        }
32        else
33        {
34            if (total < 10)
35            {
36                payment = 10;
37            }
38            else
39            {
40                payment = (10 / 100.0) * total;
41            }
42        }
43        printf("your balance : %.2f\n", balance);
44        printf("interest due : %.2f\n", interest);
45        printf("total amount due : %.2f\n", total);
46        printf("minimum payment : %.2f\n", payment);
47        printf(
48            "press q to quit or any buoton for another calculation : ");
49        char x;
50        scanf("%s", &x);
51        c = x;
52        printf("\n");
53    }while(c != 'q');
54 }
```

The output:



```
(base) alisoliman@192 iot % code 4.c
(base) alisoliman@192 iot % ./4
balance : 5000
your balance : 5000.00
interest due : 50.00
total amount due : 5050.00
minimum payement : 505.00
press q to quit or any buuton for another calculation : █
```



```
(base) alisoliman@192 iot % code 4.c
(base) alisoliman@192 iot % ./4
balance : 5000
your balance : 5000.00
interest due : 50.00
total amount due : 5050.00
minimum payement : 505.00
press q to quit or any buuton for another calculation : f

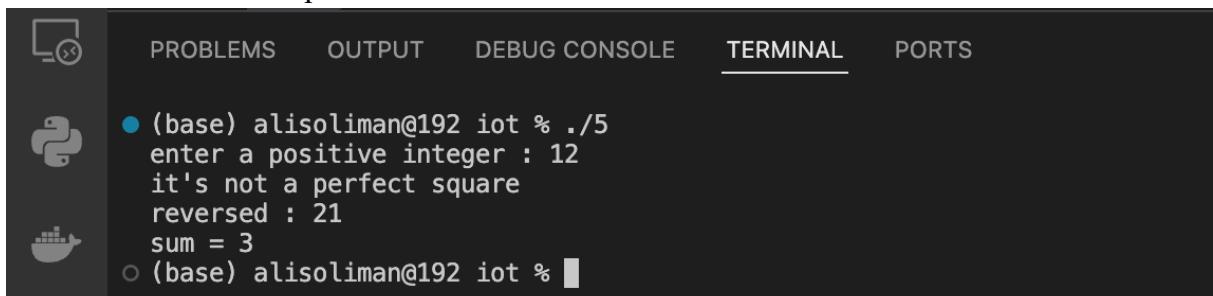
balance : 3400
your balance : 3400.00
interest due : 34.00
total amount due : 3434.00
minimum payement : 343.40
press q to quit or any buuton for another calculation : q

(base) alisoliman@192 iot % █
```

The 5th question:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void perfect_square(int num);
5 void reversed(int num);
6 int sum(int num);
7
8 int main(void)
9 {
10     int num;
11     do
12     {
13         printf("enter a positive integer : ");
14         scanf("%i", &num);
15     } while (num < 0);
16     perfect_square(num);
17     reversed(num);
18     int summ = sum(num);
19     printf("sum = %i\n", summ);
20 }
21 void reversed(int num)
22 {
23     int number = num;
24     int n = 0;
25     while (num >= 10)
26     {
27         num = num / 10;
28         n++;
29     }
30     int size = n + 1;
31     char reverse[size];
32     for int i = 0; i < size; i++
33     {
34         reverse[i] = (number % 10) + '0';
35         number = number / 10;
36     }
37     printf("reversed : ");
38     for int i = 0; i < size; i++)
39     {
40         printf("%c", reverse[i]);
41     }
42     printf("\n");
43 }
44
45 int sum(int num)
46 {
47     int sum = 0;
48     while (num >= 10)
49     {
50         sum += num % 10;
51         num = num / 10;
52     }
53     sum += num;
54     return sum;
55 }
56 void perfect_square(int num)
57 {
58     int sqrt1 = sqrt(num);
59     float sqrt2 = sqrt(num);
60     if (sqrt1 == sqrt2)
61     {
62         printf("it's a perfect square\n");
63         return;
64     }
65     printf("it's not a perfect square\n");
66 }
67 }
```

The output:

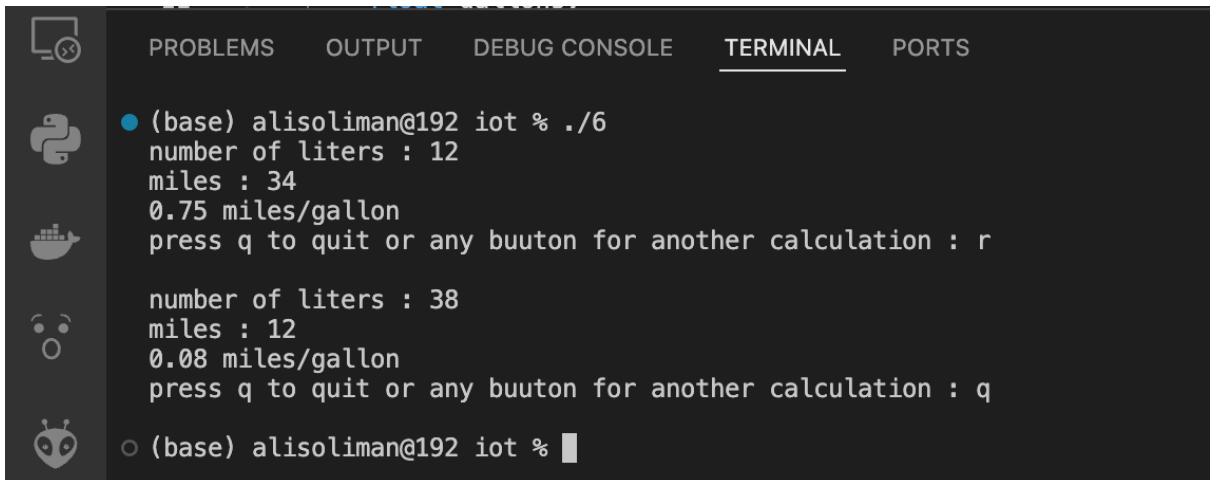


```
(base) alisoliman@192 iot % ./5
enter a positive integer : 12
it's not a perfect square
reversed : 21
sum = 3
(base) alisoliman@192 iot %
```

The 6th question:

```
1 #include <stdio.h>
2
3 const float liter = 0.264179;
4 int main(void)
5 {
6     char c = 'a';
7     do{
8         printf("number of liters : ");
9         float liters;
10        scanf("%f", &liters);
11        float gallons;
12        gallons = liters / liter;
13        printf("miles : ");
14        float mile
15        scanf("%f", &mile);
16        float m_g = miles / gallons;
17        printf("%.2f miles/gallon\n", m_g);
18        printf(
19            "press q to quit or any buoton for another calculation : ");
20        char x;
21        scanf("%s", &x);
22        c = x;
23        printf("\n");
24    }while(c != 'q');
25 }
26
```

the output:

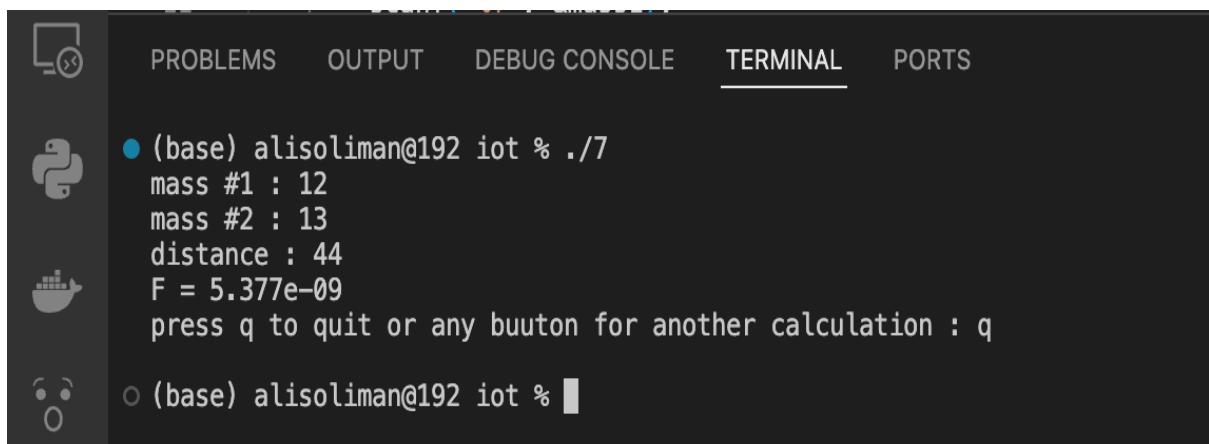


The screenshot shows a terminal window with the following interface elements:

- Top bar: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS.
- Icon bar: Python, Docker, Terminal, and a file icon.
- Terminal content:
 - Line 1: (base) alisoliman@192 iot % ./6
 - Line 2: number of liters : 12
 - Line 3: miles : 34
 - Line 4: 0.75 miles/gallon
 - Line 5: press q to quit or any buutton for another calculation : r
 - Line 6: number of liters : 38
 - Line 7: miles : 12
 - Line 8: 0.08 miles/gallon
 - Line 9: press q to quit or any buutton for another calculation : q
 - Line 10: (base) alisoliman@192 iot % █

```
1 #include<stdio.h>
2 #include <math.h>
3
4 const double G = 6.673e-8;
5 int main(void)
6 {
7     char c = 'a';
8     do{
9         printf("mass #1 : ");
10        float mass
11        scanf("%f", &mass1);
12        printf("mass #2 : ");
13        float mass
14        scanf("%f", &mass2);
15        printf("distance : ");
16        float dis;
17        scanf("%f", &dis);
18        float F;
19        F = (G * mass1 * mass2) / pow(dis, 2);
20        printf("F = %g\n", F);
21        printf(
22            "press q to quit or any buuton for another calculation : "
23        );
24        char x;
25        scanf("%s", &x);
26        c = x;
27    }while( c != 'q');
28 }
```

The output:



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● (base) alisoliman@192 iot % ./7
mass #1 : 12
mass #2 : 13
distance : 44
F = 5.377e-09
press q to quit or any buuton for another calculation : q

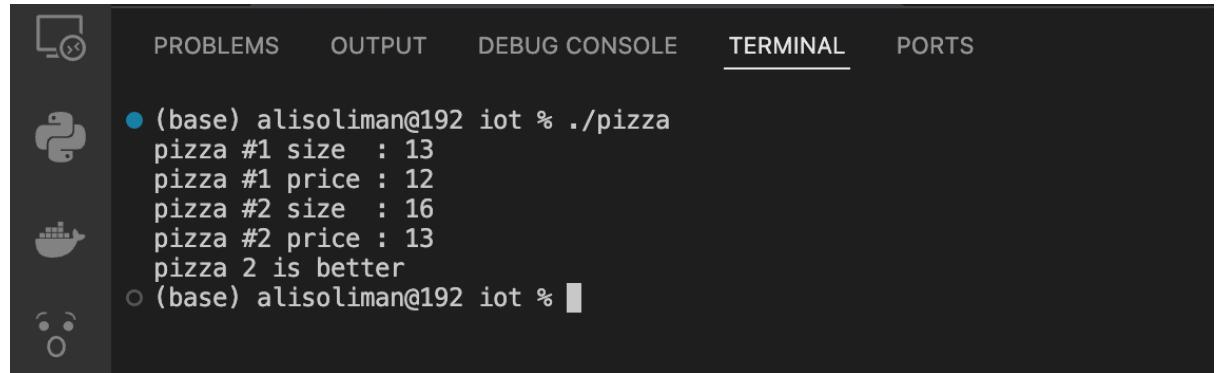
○ (base) alisoliman@192 iot % █

```

1  #include <stdio.h>
2  #include<ctype.h>
3
4  float price_in(int diameter, int price);
5
6  int main(void)
7  {
8      typedef struct
9  {
10      float size;
11      float pric
12      float tota
13 }pizza; l;
14
15     char s = '0', p = '0';
16     pizza prices[2];
17     for int i = 0; i < 2; i++)
18     {
19         printf("pizza #%i size : ",(i + 1));
20         scanf("%f", &prices[i].size);
21         printf("pizza #%i price : ",(i + 1));
22         scanf("%f", &prices[i].price);
23         prices[i].total = price_in(prices[i].size, prices
24 [i].price);
25     }
26     if (prices[0].total > prices[1].total)
27     {
28         printf("pizza 2 is better\n");
29         return 0;
30     }
31     if (prices[0].total == prices[1].total)
32     {
33         if (prices[0].size < prices[1].size)
34         {
35             printf("pizza 1 is better\n");
36             return 0;
37         }
38         if (prices[0].size < prices[1].size)
39         {
40             printf("pizza 2 is better\n");
41             return 0;
42         }
43         else
44         {
45             printf("they're both the same\n");
46             return 1;
47         }
48     }
49     else
50     {
51         printf("pizza 1 is better\n");
52         return 0;
53     }
54 }
55 float price_in(int diameter, int pric
56 {                           e)
57     float r = diameter / 2;
58     float area = 3.14 * r * r;
59     float total = price / area;
60     return tota
61 }                           l;
62

```

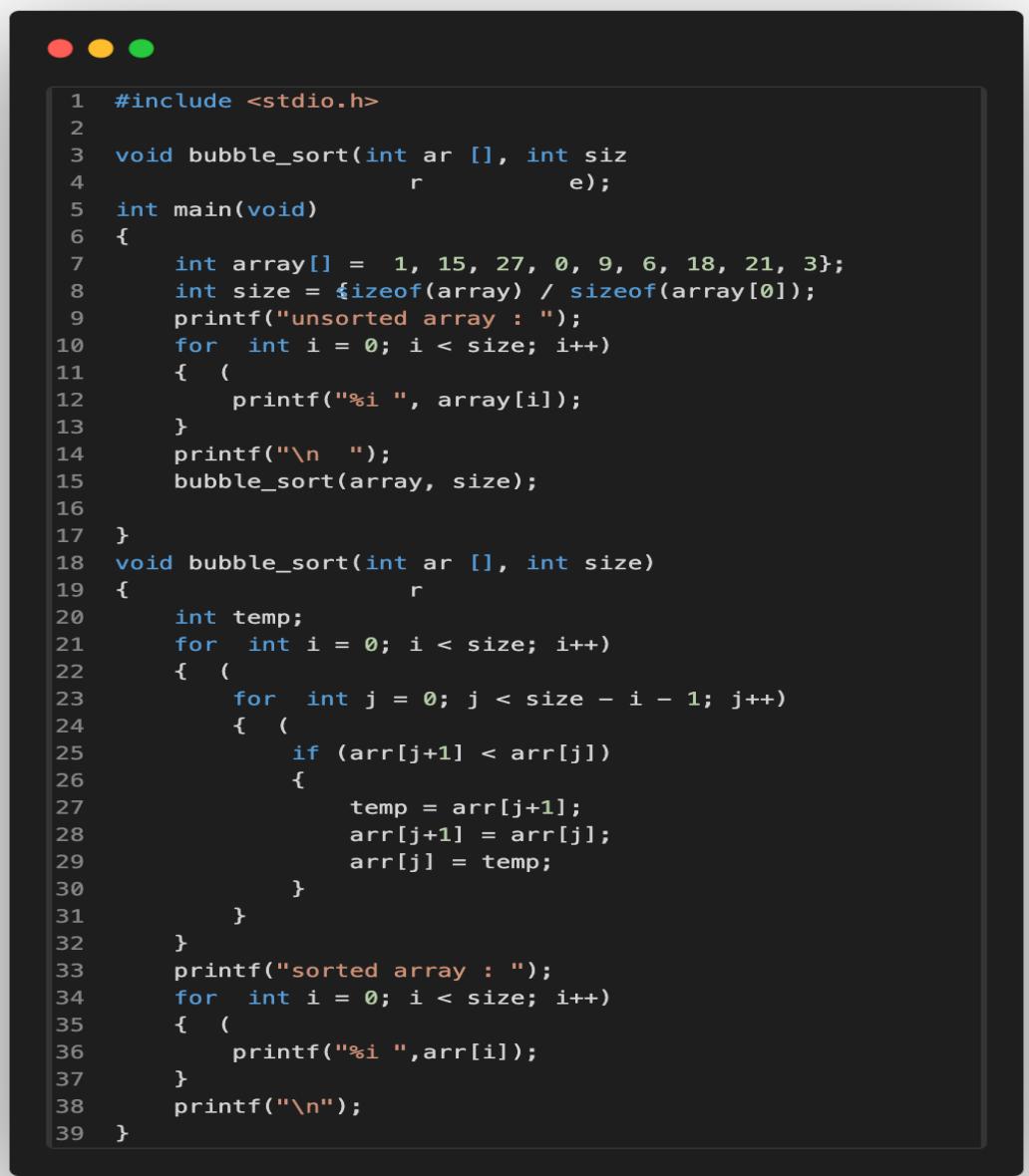
The output:



The terminal window shows the following output:

```
(base) alisoliman@192 iot % ./pizza
pizza #1 size : 13
pizza #1 price : 12
pizza #2 size : 16
pizza #2 price : 13
pizza 2 is better
(base) alisoliman@192 iot %
```

The double bonus question:



```
1 #include <stdio.h>
2
3 void bubble_sort(int arr[], int size)
4 {
5     int array[] = {1, 15, 27, 0, 9, 6, 18, 21, 3};
6     int size = sizeof(array) / sizeof(array[0]);
7     printf("unsorted array : ");
8     for (int i = 0; i < size; i++)
9     {
10         printf("%i ", array[i]);
11     }
12     printf("\n");
13     bubble_sort(array, size);
14 }
15
16
17 }
18 void bubble_sort(int arr[], int size)
19 {
20     int temp;
21     for (int i = 0; i < size; i++)
22     {
23         for (int j = 0; j < size - i - 1; j++)
24         {
25             if (arr[j+1] < arr[j])
26             {
27                 temp = arr[j+1];
28                 arr[j+1] = arr[j];
29                 arr[j] = temp;
30             }
31         }
32     }
33     printf("sorted array : ");
34     for (int i = 0; i < size; i++)
35     {
36         printf("%i ", arr[i]);
37     }
38     printf("\n");
39 }
```

The output:

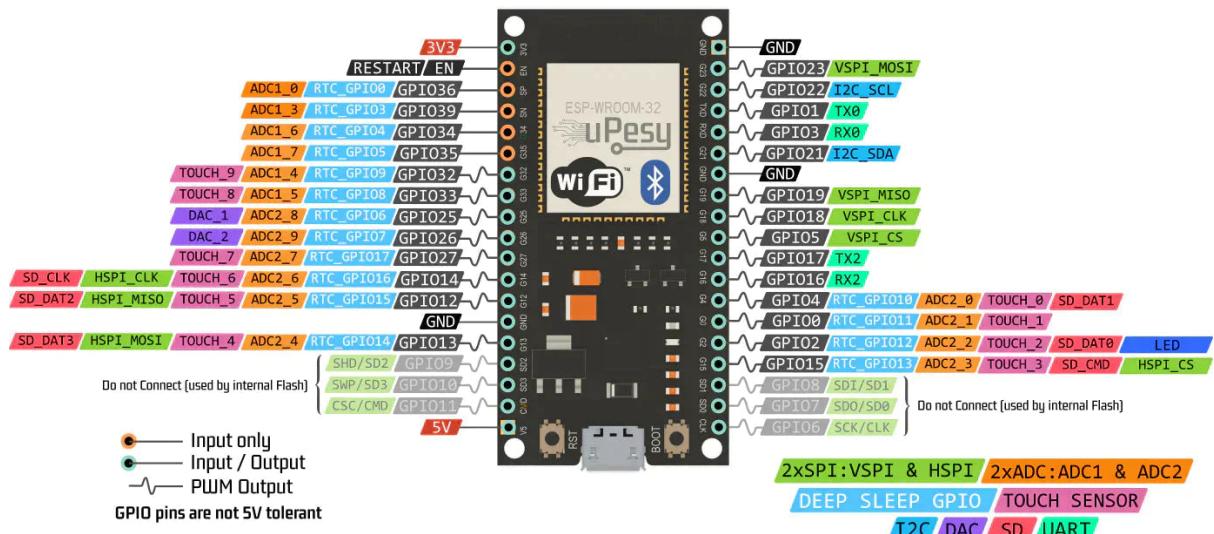
```
● (base) alisoliman@192 iot % code bubble_sort.c
● (base) alisoliman@192 iot % ./bubble_sort
    unsorted array : 1 15 27 0 9 6 18 21 3
    sorted array : 0 1 3 6 9 15 18 21 27
○ (base) alisoliman@192 iot %
```

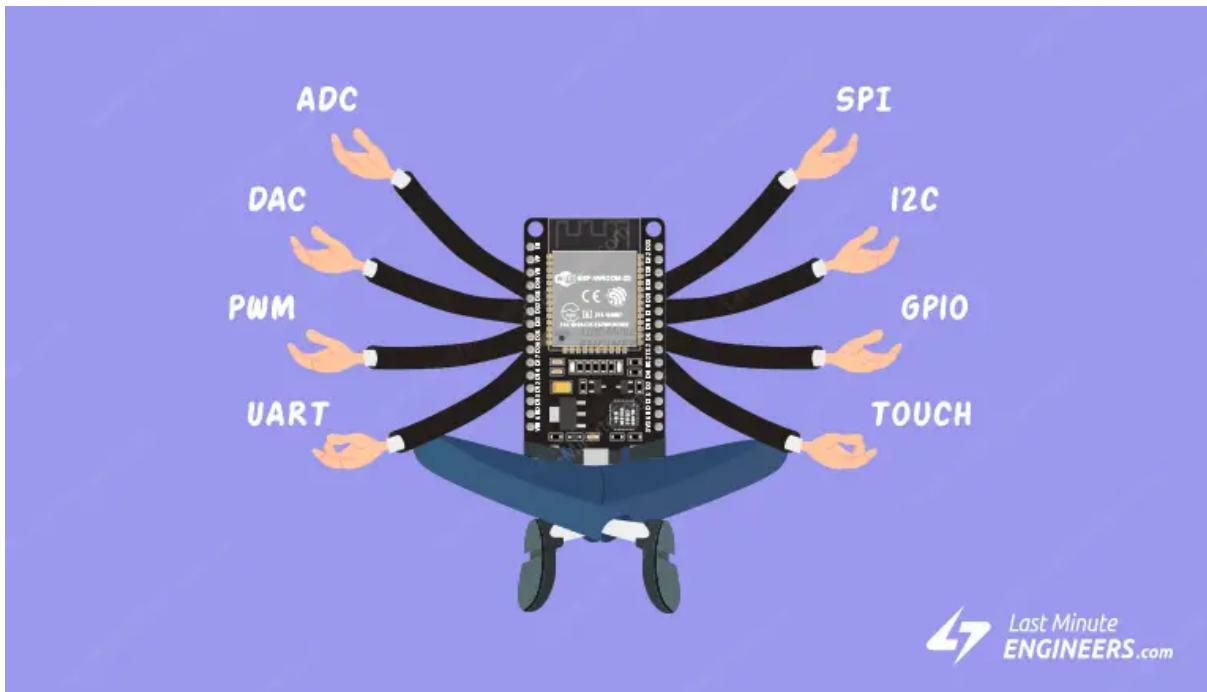
x. I've worked on the 1st theoretical question also.

1. ESP32 based Embedded System:

- i. 40 to 60 hours.
- ii. DR/Mohamed Hatem Eng/ fady Eng/Mahmoud Essam
- iii. Well by that chapter I had no background what so ever about hardware micro controllers (ESP32) or any of the hardware pieces include in the kit and so the outcome was that I learned everything I needed to know about all of those how to use and control them and how to connect them to the esp32 and I learned all about each of the hardware pieces and which should be connected digital and analog.
- iv. The theoretical parts included the micro controllers and the esp32 in particular what are each of the hardware pieces what they're used for and how.
- v.

ESP32 Wroom DevKit Full Pinout





- vi. I worked on the onsite question which was to connect both of the servo motor and the keypad to the esp32 and assign some of the buttons on the keypad to control the servo's movement.
- vii.

Screenshot of the PlatformIO IDE showing the code for a project named "servo_keypad". The code is written in C++ and includes headers for Arduino, Keypad, and ESP32Servo. It defines a 4x4 keypad keymap and initializes pins for the keypad and a servo. The setup() function initializes serial communication and attaches the servo to pin 15. The loop() function reads keypad input and prints it to the serial port.

```

#include <Arduino.h>
#include <Keypad.h>
#include<ESP32Servo.h>
#include <iostream>

// Define the number of rows and columns in the keypad
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Four columns

Servo myservo;

// Define the keymap
char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

// Connect keypad ROW1, ROW2, ROW3, ROW4 to these pins
byte rowPins[ROWS] = {13, 12, 14, 27};

// Connect keypad COL1, COL2, COL3, COL4 to these pins
byte colPins[COLS] = {26, 25, 33, 32};

// Create the Keypad object
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
    Serial.begin(9600); // Initialize serial communication
    myservo.attach(15);
    myservo.write(0);
    Serial.print("A.30 B.45 C.60 D.90 #.120 *.180 0.entry : ");
    pinMode(5, INPUT);
}

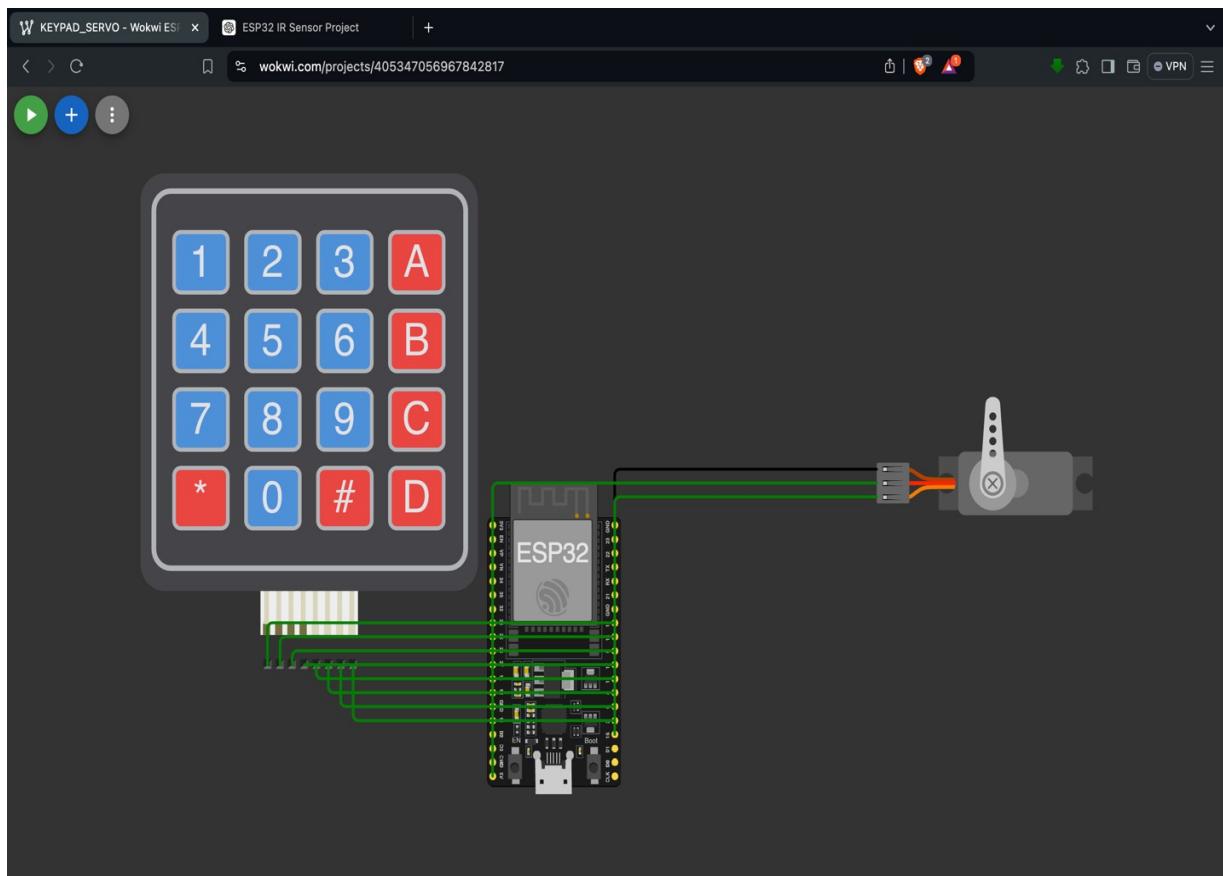
void loop() {
    int irread;
    irread = digitalRead(5);
    char key = keypad.getKey(); // Get the key pressed
    if (key) {
        Serial.println(key);
    }
}

```

The screenshot shows the PlatformIO IDE interface. The left sidebar displays the project structure under 'UNTITLED (WORKSPACE)'. The 'src' folder contains the file 'main.cpp', which is currently selected and shown in the main code editor area. The code in 'main.cpp' is as follows:

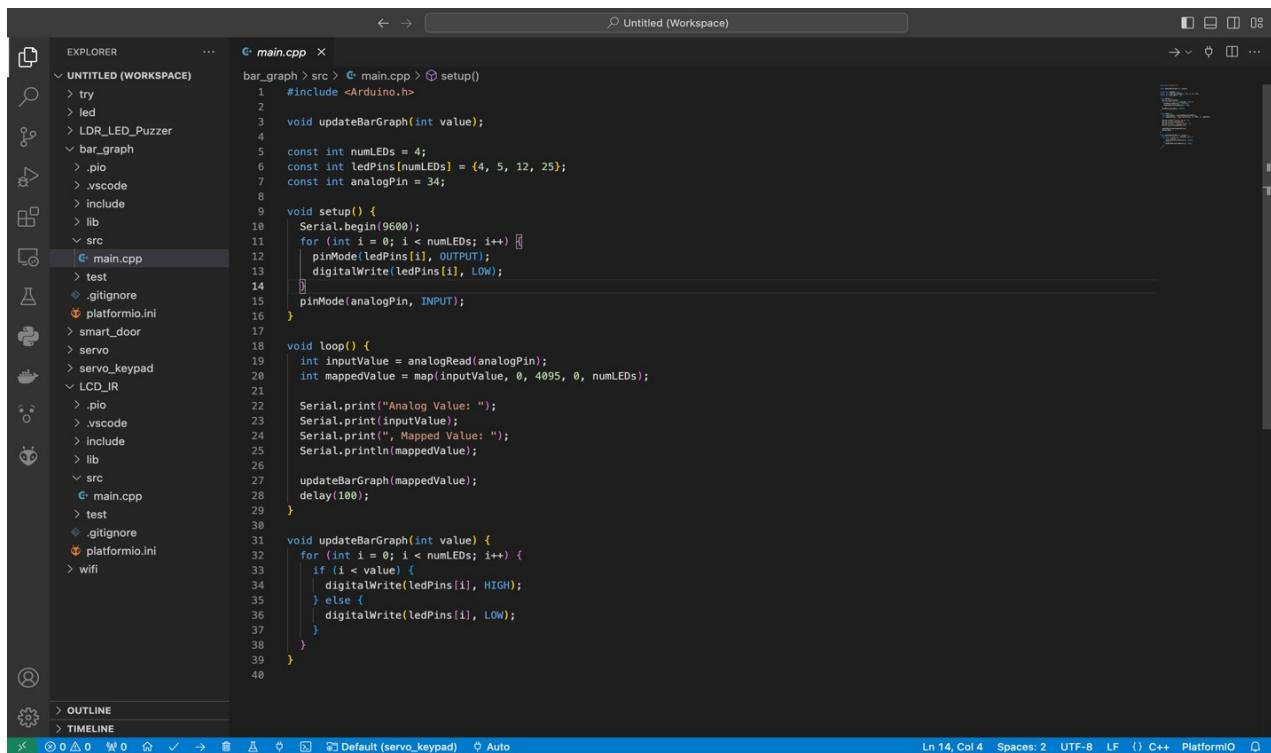
```
servo_keypad > src > main.cpp > ...
38 void loop() {
39     int irread;
40     irread = digitalRead(5);
41     char key = keypad.getKey(); // Get the key pressed
42     if (key)
43     {
44         Serial.println(key);
45     }
46     switch (key)
47     {
48         case 'A':
49             myservo.write(30);
50             break;
51         case 'B':
52             myservo.write(45);
53             break;
54         case 'C':
55             myservo.write(60);
56             break;
57         case 'D':
58             myservo.write(90);
59             break;
60         case '#':
61             myservo.write(120);
62             break;
63         case '*':
64             myservo.write(180);
65             break;
66         case '0':
67             Serial.println("Enter a number:"); // Prompt the user for input
68             while (Serial.available() == 0)
69             {
69                 // Wait for input
70             }
71             float number = Serial.parseFloat(); // Read the float value
72             myservo.write(number); // Write the value to the servo
73             Serial.print("Servo angle set to: ");
74             Serial.println(number); // Print the value
75             break;
76         }
77     if (irread == 0)
78     {
79         myservo.write(0);
80     }
81 }
```

The status bar at the bottom indicates the current line (Ln 19, Col 1), spaces (Spaces: 2), and encoding (UTF-8). The right side of the interface includes a terminal window and various status indicators.

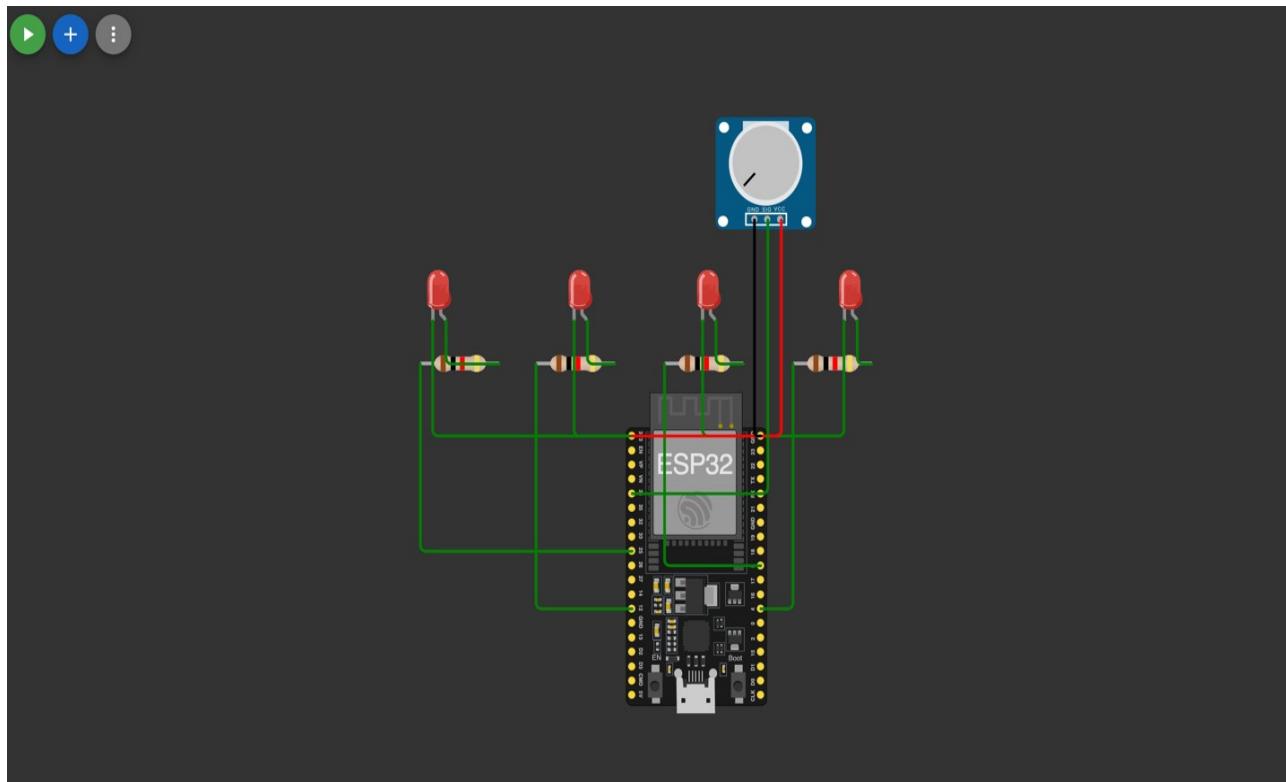


viii. I've worked on all the esp32 questions 1,2,3,bonus and problem 1,2,3.

ix. 1st question:



```
bar_graph > src > main.cpp > setup()
1 #include <Arduino.h>
2
3 void updateBarGraph(int value);
4
5 const int numLEDs = 4;
6 const int ledPins[numLEDs] = {4, 5, 12, 25};
7 const int analogPin = 34;
8
9 void setup() {
10     Serial.begin(9600);
11     for (int i = 0; i < numLEDs; i++) {
12         pinMode(ledPins[i], OUTPUT);
13         digitalWrite(ledPins[i], LOW);
14     }
15     pinMode(analogPin, INPUT);
16 }
17
18 void loop() {
19     int inputValue = analogRead(analogPin);
20     int mappedValue = map(inputValue, 0, 4095, 0, numLEDs);
21
22     Serial.print("Analog Value: ");
23     Serial.print(inputValue);
24     Serial.print(", Mapped Value: ");
25     Serial.println(mappedValue);
26
27     updateBarGraph(mappedValue);
28     delay(100);
29 }
30
31 void updateBarGraph(int value) {
32     for (int i = 0; i < numLEDs; i++) {
33         if (i < value) {
34             digitalWrite(ledPins[i], HIGH);
35         } else {
36             digitalWrite(ledPins[i], LOW);
37         }
38     }
39 }
```



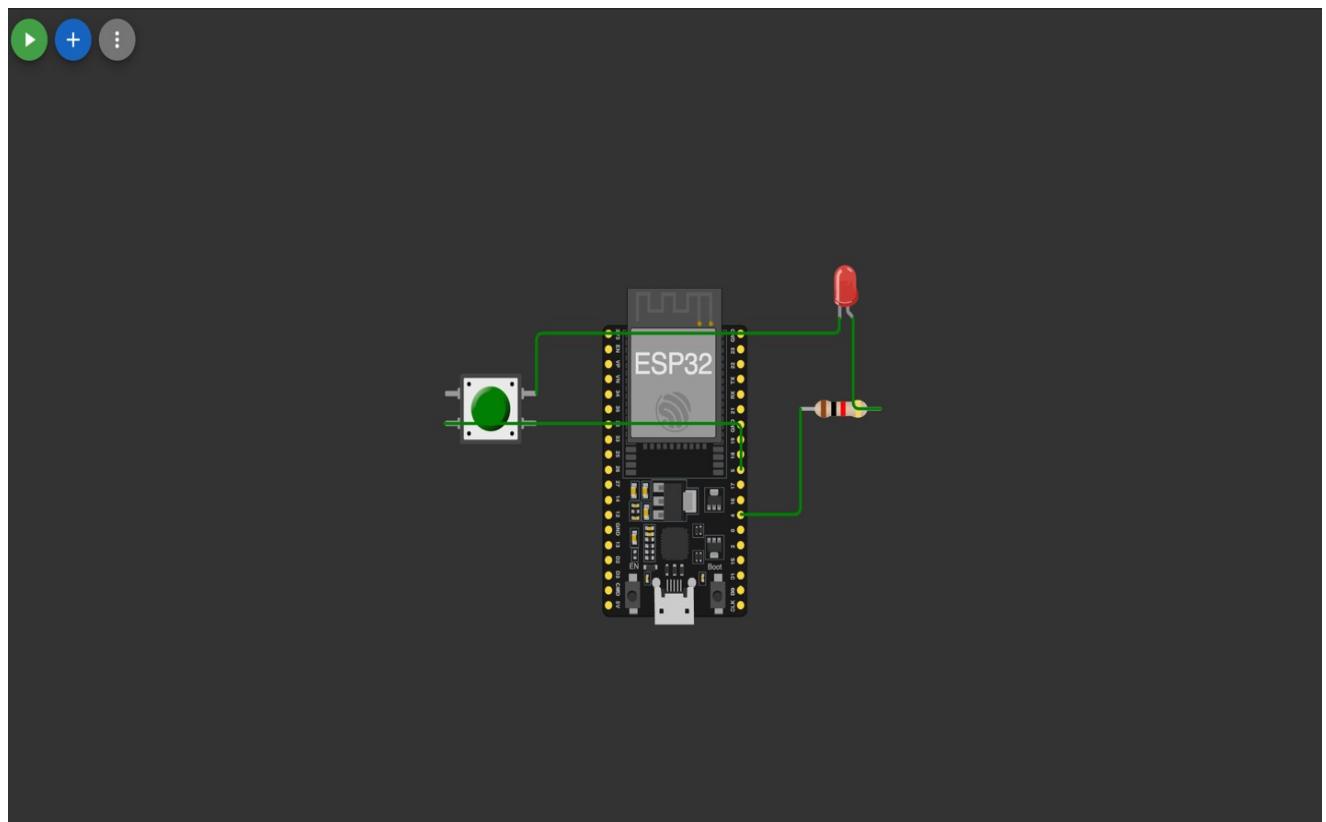
2nd question:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER**: Shows the project structure under "UNTITLED (WORKSPACE)". The "main.cpp" file is selected.
- EDITOR**: Displays the content of "main.cpp". The code is as follows:

```
led > src > main.cpp > loop()
1 #include <Arduino.h>
2
3 // put function declarations here:
4
5 void setup() {
6     // put your setup code here, to run once:
7     pinMode(4, OUTPUT);
8     Serial.begin(9600);
9     pinMode(5, INPUT);
10    digitalWrite(4,LOW);
11 }
12
13 void loop() {
14     // put your main code here, to run repeatedly:
15     int irread ;
16     irread = digitalRead(5);
17
18     if (irread == 0) {
19         Serial.println("button pressed");
20         digitalWrite(4, HIGH);
21     }
22     else {
23         Serial.println("button not pressed");
24         digitalWrite(4, LOW);
25     }
26 }
27
28 }
```

- STATUS BAR**: Shows "Ln 17, Col 28" and "Spaces: 2" and "LF".

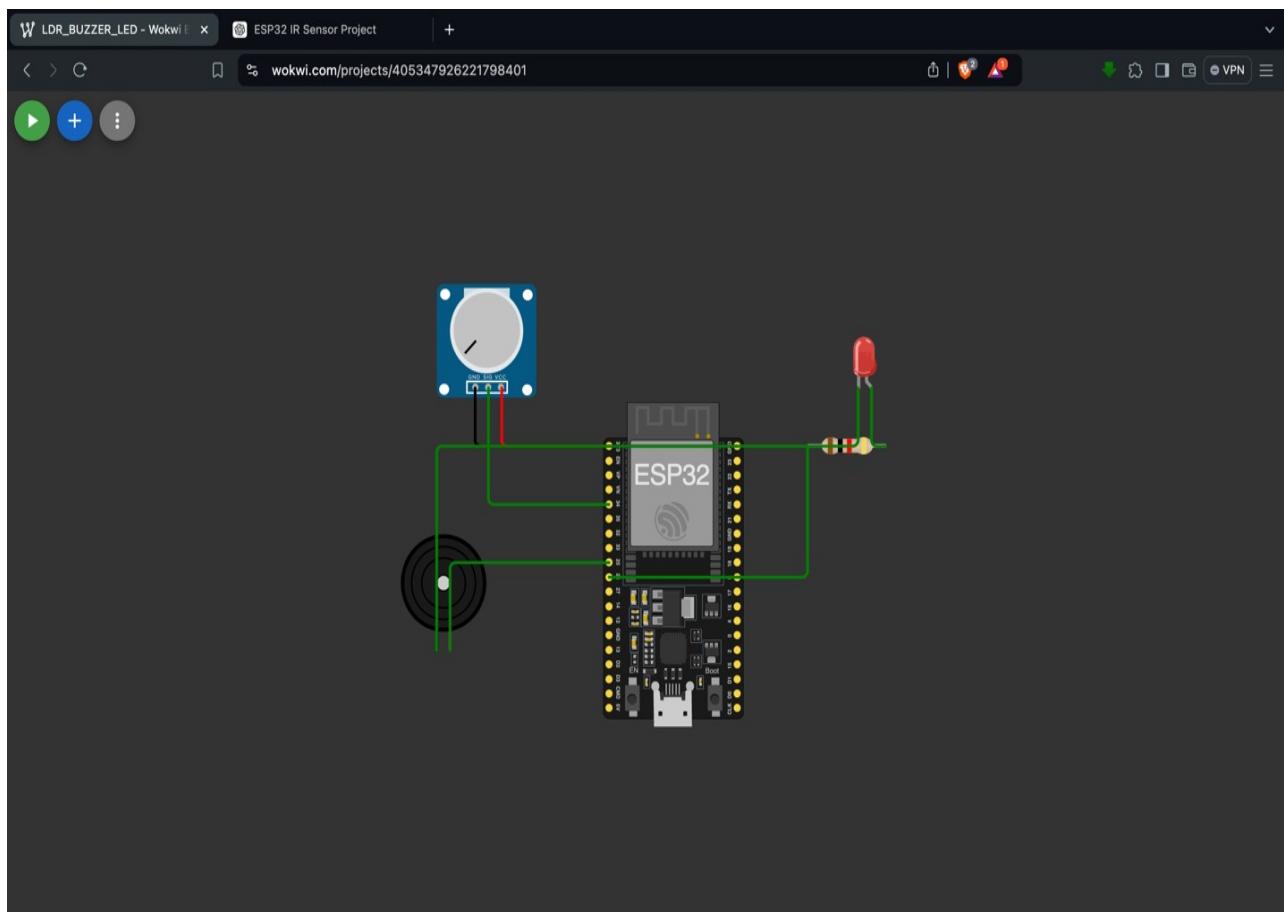


3rd question:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure for "UNTITLED (W...)". The "main.cpp" file is selected.
- Code Editor:** Displays the C++ code for "main.cpp". The code reads analog values from an LDR connected to pin 36, prints them to the Serial Monitor, and controls a buzzer connected to pin 4 based on the LDR value.
- Bottom Status Bar:** Shows "Ln 5, Col 18" and "Spaces: 2" settings.

```
#include <Arduino.h>
#define LDR_PIN 36 // ESP32 pin connected to the LDR
#define buzzer_PIN 4
#define LED_PIN 5
void setup() {
    Serial.begin(9600); // Initialize serial communication
    pinMode(LED_PIN, OUTPUT);
}
void loop() {
    int ldrValue = analogRead(LDR_PIN); // Read the analog value from the LDR
    Serial.print("LDR Value: ");
    Serial.println(ldrValue); // Print the value to the Serial Monitor
    if (ldrValue < 300)
    {
        tone(buzzer_PIN, 100);
        digitalWrite(LED_PIN, HIGH);
        delay(100);
        noTone(buzzer_PIN);
        digitalWrite(LED_PIN, LOW);
        delay(100);
    }
    delay(100);
}
```



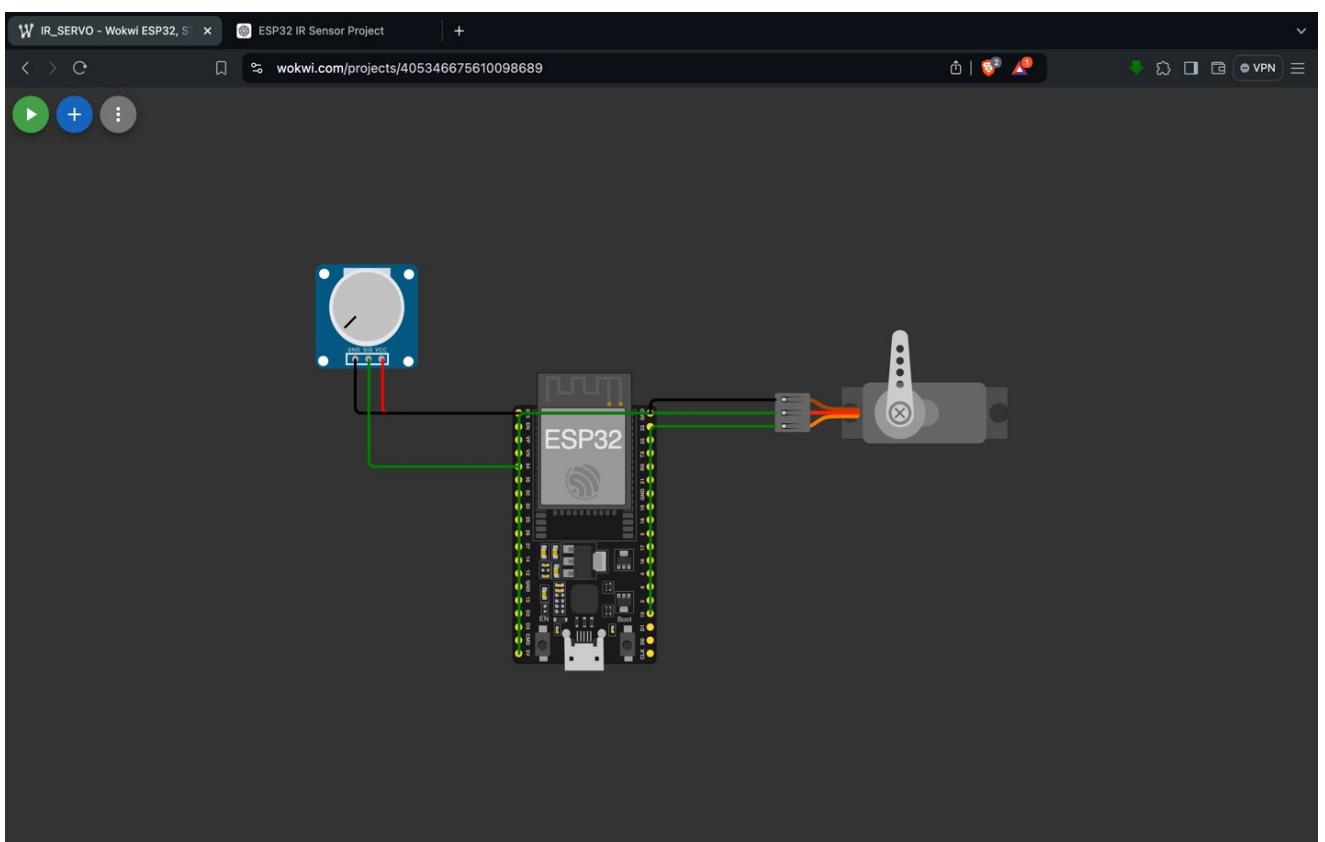
4th question:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** view: Shows the project structure under "UNTITLED (WORKSPACE)". The "main.cpp" file is currently selected.
- CODE** view: Displays the content of the "main.cpp" file. The code is as follows:

```
smart_door > src > main.cpp > loop()
1 #include <Arduino.h>
2 #include <ESP32Servo.h>
3
4 const int irPin = 4;      // GPIO pin connected to the IR sensor's output
5 const int servoPin = 15;  // GPIO pin connected to the servo motor's signal
6
7 Servo myservo;           // Create a servo object
8
9 void setup() {
10    Serial.begin(9600);    // Start serial communication for debugging
11    pinMode(irPin, INPUT); // Set the IR sensor pin as an input
12    myservo.attach(servoPin); // Attach the servo to the specified pin
13    myservo.write(0);       // Initialize servo to 0 degrees
14 }
15
16 void loop() {
17    int irValue = digitalRead(irPin); // Read the digital value from the IR sensor
18
19    if (irValue == LOW) {           // If an object is detected
20        myservo.write(180);        // Move the servo to 90 degrees
21        Serial.println("someone is at the door: Servo at 180 degrees");
22    } else {
23        myservo.write(0);          // Move the servo to 0 degrees
24        Serial.println("noone is at the door: Servo at 0 degrees");
25    }
26    delay(100);
27    myservo.write(0);
28
29 }
```

The status bar at the bottom indicates: Ln 16, Col 14 Spaces: 2 UTF-8 LF () C++ PlatformIO

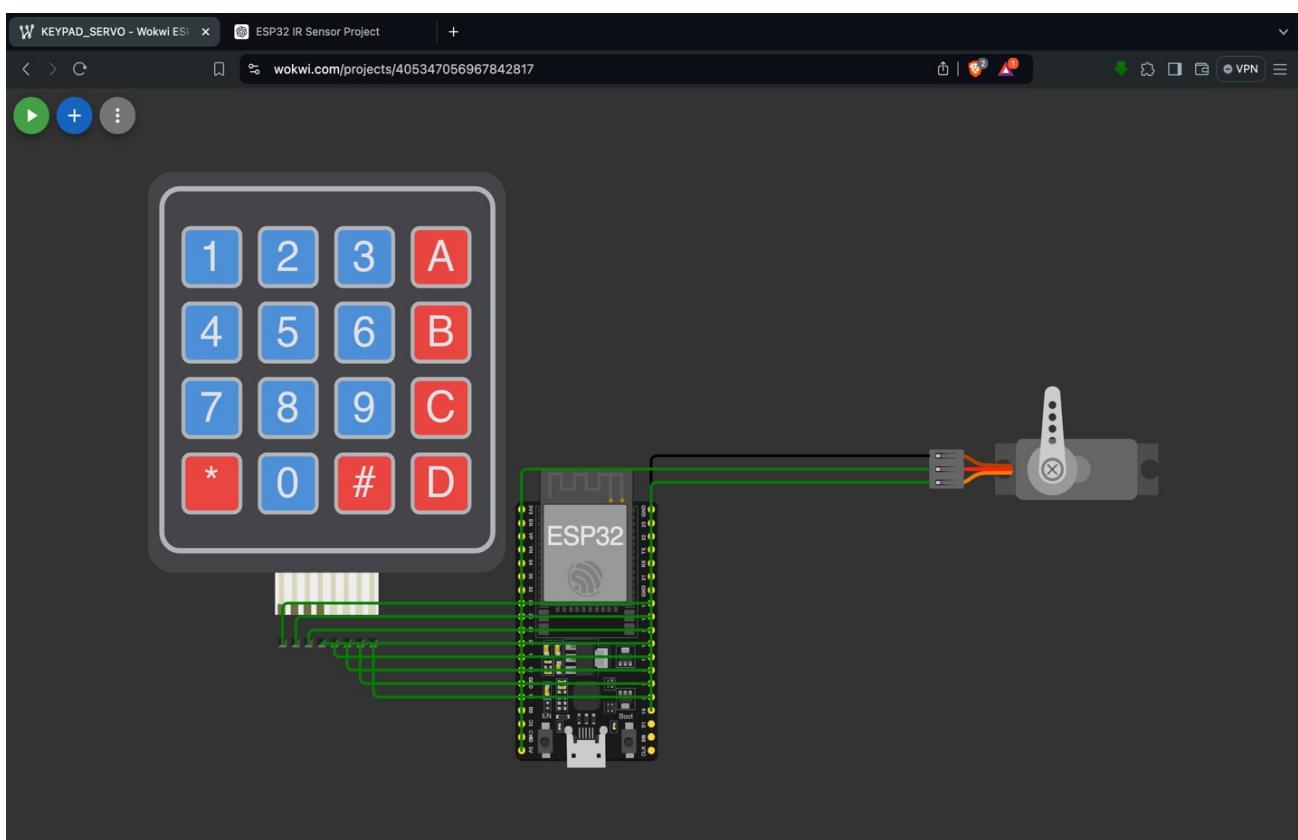


5th question:

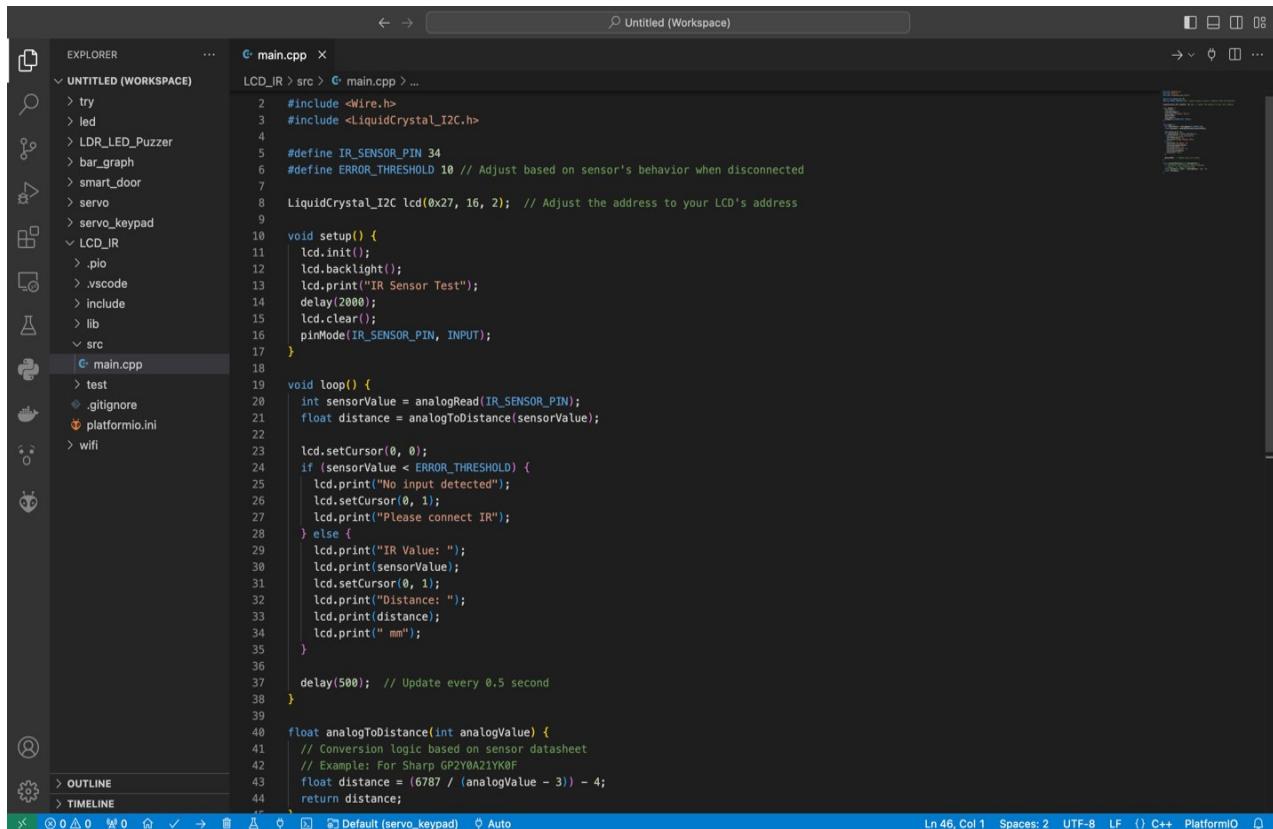
The screenshot shows the PlatformIO IDE interface. The left sidebar displays the project structure under 'EXPLORER'. The 'src' folder contains the file 'main.cpp', which is currently selected and shown in the main code editor area. The code in 'main.cpp' is as follows:

```
servo_keypad > src > main.cpp > ...
38 void loop() {
39     int irread;
40     char key = keypad.getKey(); // Get the key pressed
41     if (key)
42     {
43         Serial.println(key);
44     }
45     switch (key)
46     {
47         case 'A':
48             myservo.write(30);
49             break;
50         case 'B':
51             myservo.write(45);
52             break;
53         case 'C':
54             myservo.write(60);
55             break;
56         case 'D':
57             myservo.write(90);
58             break;
59         case '#':
60             myservo.write(120);
61             break;
62         case '*':
63             myservo.write(180);
64             break;
65         case '0':
66             Serial.println("Enter a number:"); // Prompt the user for input
67             while (Serial.available() == 0)
68             {
69                 // Wait for input
70             }
71             float number = Serial.parseFloat(); // Read the float value
72             myservo.write(number); // Write the value to the servo
73             Serial.print("Servo angle set to: ");
74             Serial.println(number); // Print the value
75             break;
76         default:
77             if (irread == 0)
78             {
79                 myservo.write(0);
80             }
81     }
}
```

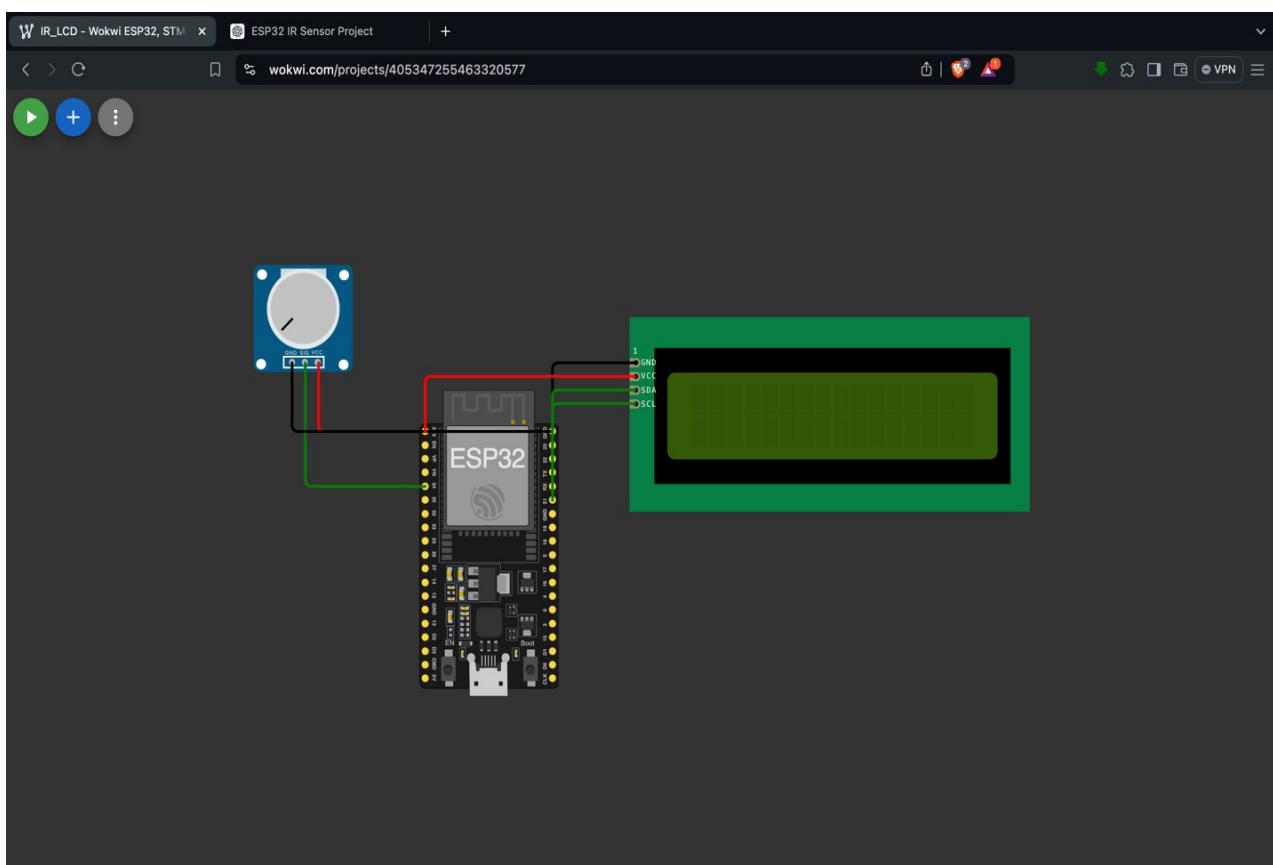
The status bar at the bottom indicates the current line (Ln 19, Col 1), spaces (Spaces: 2), encoding (UTF-8), and file type (C++). A message 'Default (servo_keypad)' is also present.



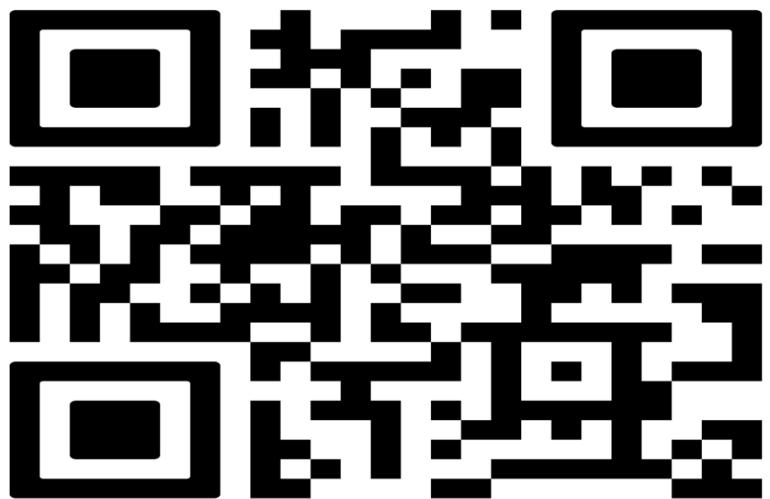
6th question:



```
main.cpp
1 //include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4
5 #define IR_SENSOR_PIN 34
6 #define ERROR_THRESHOLD 10 // Adjust based on sensor's behavior when disconnected
7
8 LiquidCrystal_I2C lcd(0x27, 16, 2); // Adjust the address to your LCD's address
9
10 void setup() {
11     lcd.init();
12     lcd.backlight();
13     lcd.print("IR Sensor Test");
14     delay(2000);
15     lcd.clear();
16     pinMode(IR_SENSOR_PIN, INPUT);
17 }
18
19 void loop() {
20     int sensorValue = analogRead(IR_SENSOR_PIN);
21     float distance = analogToDistance(sensorValue);
22
23     lcd.setCursor(0, 0);
24     if (sensorValue < ERROR_THRESHOLD) {
25         lcd.print("No input detected");
26         lcd.setCursor(0, 1);
27         lcd.print("Please connect IR");
28     } else {
29         lcd.print("IR Value: ");
30         lcd.print(sensorValue);
31         lcd.setCursor(0, 1);
32         lcd.print("Distance: ");
33         lcd.print(distance);
34         lcd.print(" mm");
35     }
36     delay(500); // Update every 0.5 second
37 }
38
39 float analogToDistance(int analogValue) {
40     // Conversion logic based on sensor datasheet
41     // Example: For Sharp GP2Y0A21YK0F
42     float distance = (6787 / (analogValue - 3)) - 4;
43     return distance;
44 }
```



The Link to the videos and all:



scaw me

- x. I've worked on any of the theoretical questions in the assignment but I've done my research about each of the topics included all over.

2. Networks & Cloud Computing using Firebase & HIVEMQ

i. 40 to 60 hours maybe more.

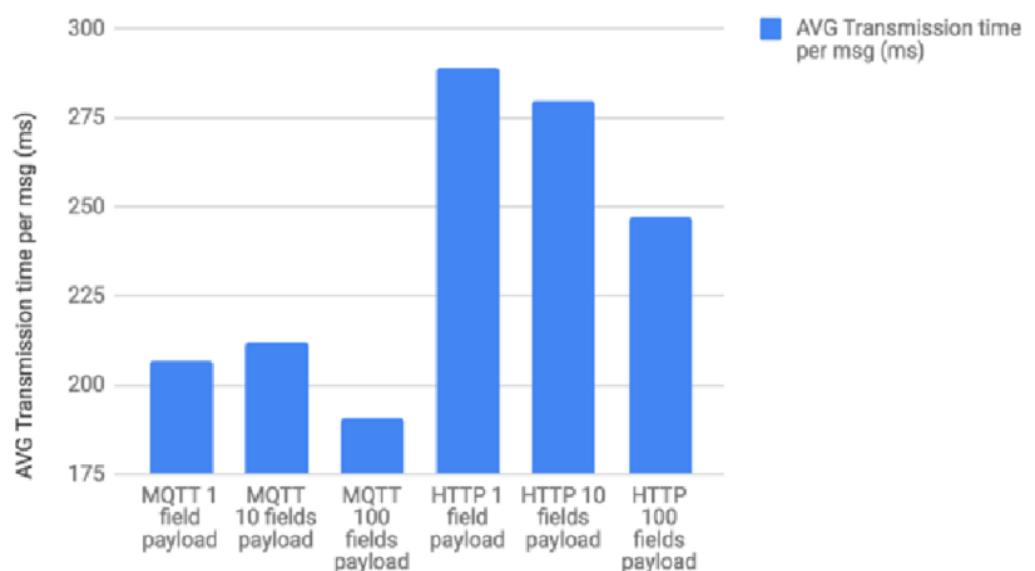
ii. Dr/ Mohamed Hatem. Eng/ Fady. Eng/ Mahmoud Essam.

iii. Well by that chapter I had a decent background of hardware connections and schematic and how to use the esp32 and how to connect each of the pieces included in the kit we purchased to it and control them but I had no background what so ever of cloud HIVEMQ and what they're used for ... so the outcome was that I learn what they are what they are used for why and when.

iv.

| MQTT | HTTP |
|---|--|
| Publish / Subscribe architecture. | Request / Response architecture. |
| Designed specifically for the use of two or more IoT devices. | HTTP protocol is to provide any document (Text, Json, Image or Video) available across the internet. |
| MQTT is bidirectional | HTTP is unidirectional, meaning the server only responds to the request from the client. |
| There can be "N" numbers of messages that can be sent in both directions. | HTTP deals with one request one at a time |
| In MQTT topics are points of target. | In HTTP URIs , we can say point of target. |
| At a time 256MB message size is maximum. | Usually there is no limit for HTTP protocol. |
| Messages can be distributed from One broker to many clients. | One to One message distribution. |

Average transmission time per message (ms)



v.In the session we've all encountered the same error so we went home with the same error until it was wolved in the online session.

vi.They'll be included on the assignment.

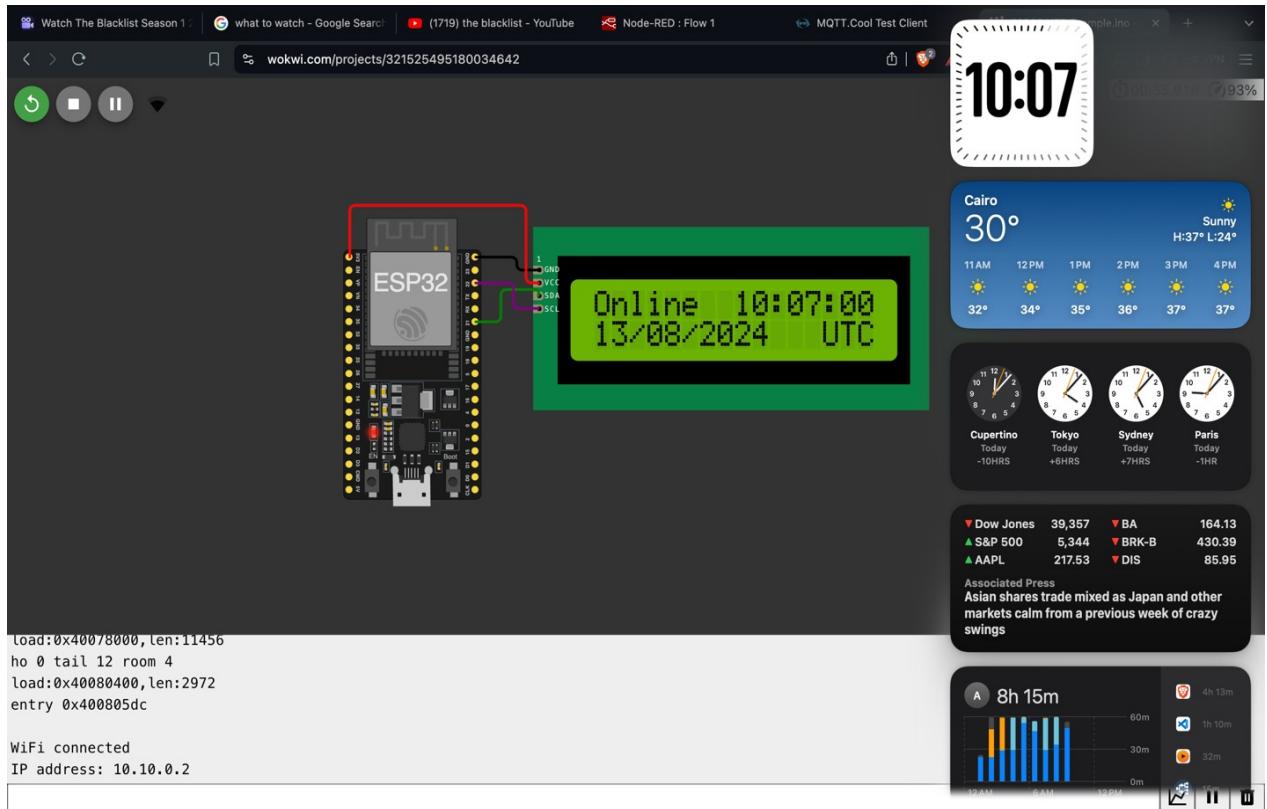
vii.

viii.I've worked on all the questions regarding the global esp32 in this assignment s0 1,2,3,4,5.

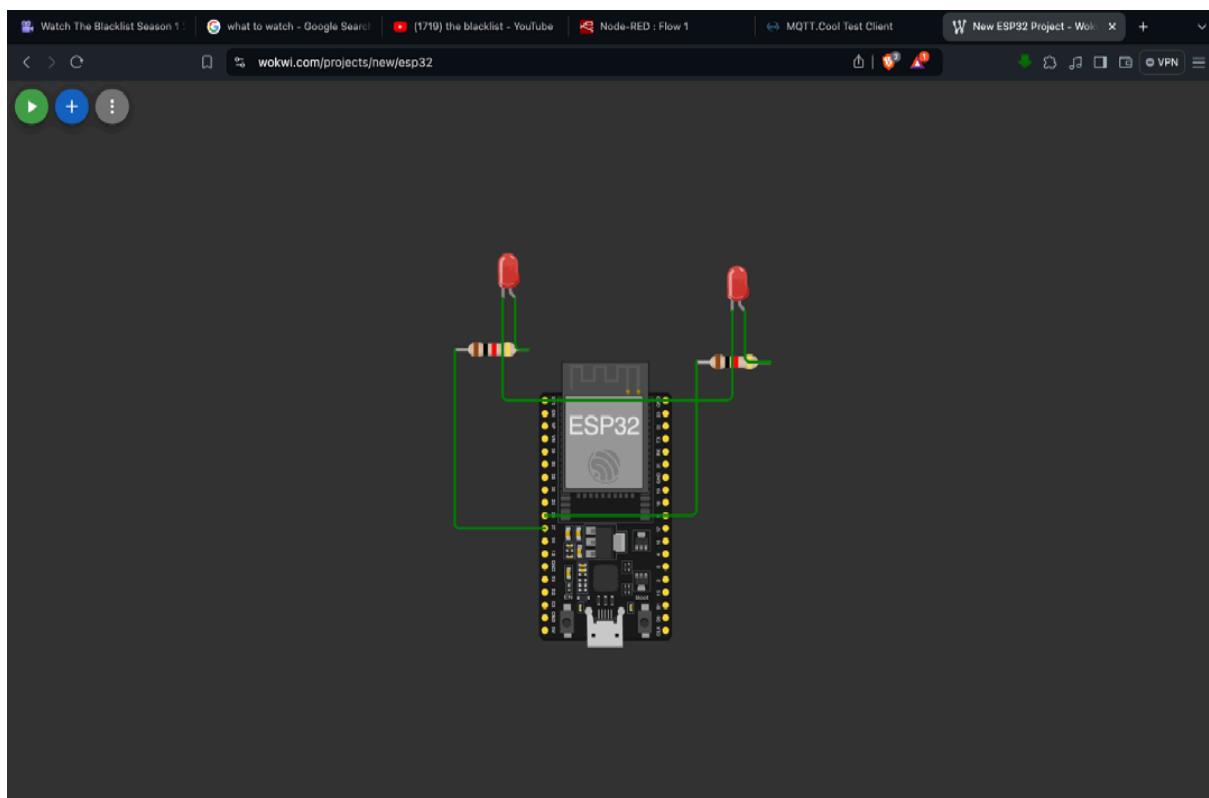


```
1 #include <WiFi.h>
2 #include <NTPClient.h>
3 #include <WiFiUdp.h>
4
5 // Replace with your network credentials
6 const char* ssid = "WE_9BFF74";
7 const char* password = "*****";
8
9 // NTP Server
10 const char* ntpServer = "pool.ntp.org";
11 const long gmtOffset_sec = 10800;
12 const int daylightOffset_sec = 3600;
13
14 // Define NTP Client to get time
15 WiFiUDP ntpUDP;
16 NTPClient timeClient(ntpUDP, ntpServer, gmtOffset_sec,
17 60000);
18
19 void setup() {
20     // Initialize Serial Monitor
21     Serial.begin(9600);
22
23     // Connect to Wi-Fi
24     Serial.print("Connecting to ");
25     Serial.println(ssid);
26     WiFi.begin(ssid, password);
27     while (WiFi.status() != WL_CONNECTED) {
28         delay(1000);
29         Serial.println("Connecting...");
30     }
31     Serial.println("Connected to Wi-Fi");
32
33     // Initialize a NTPClient to get time
34     timeClient.begin();
35 }
```

ix. 1st question:

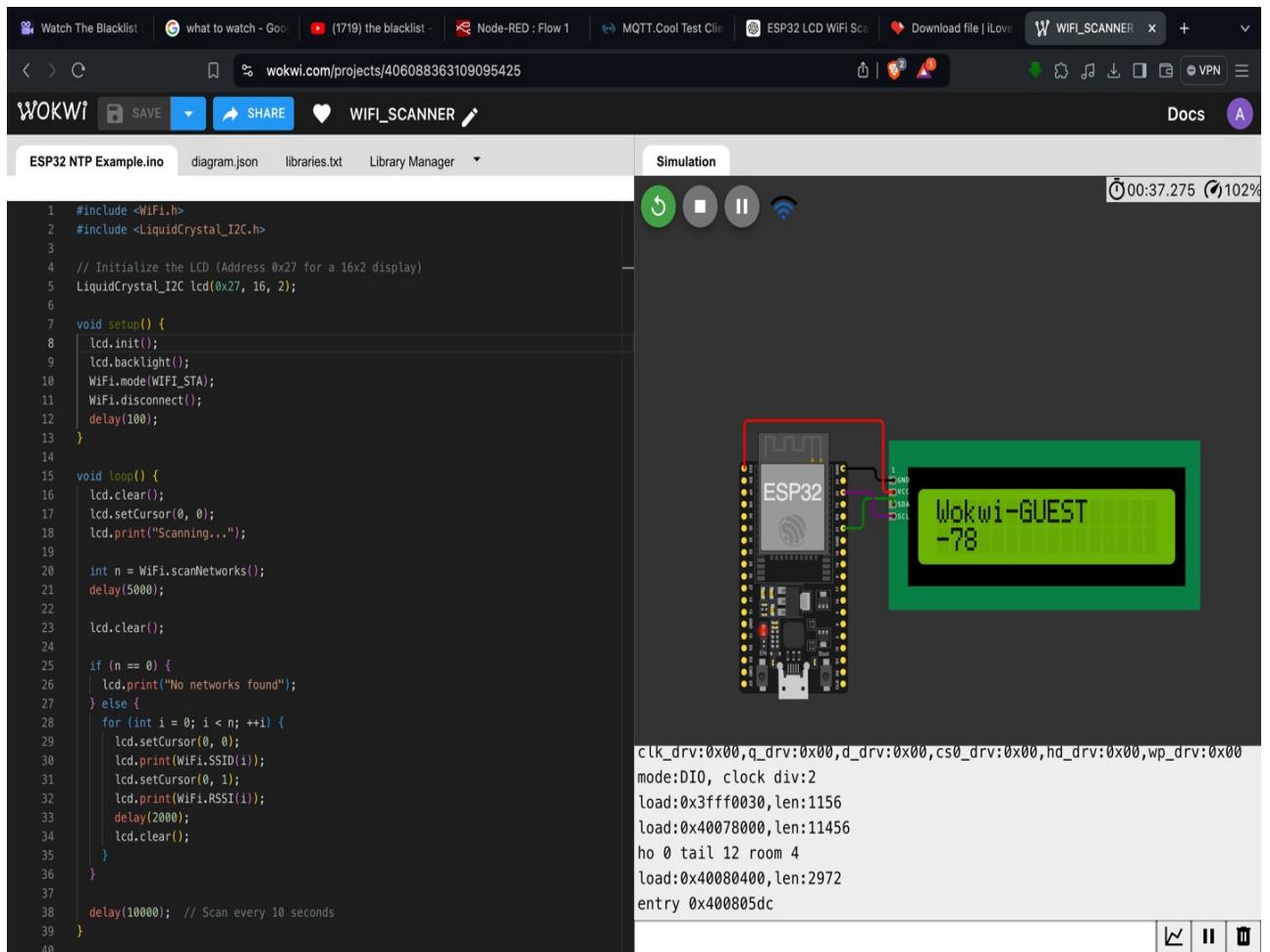
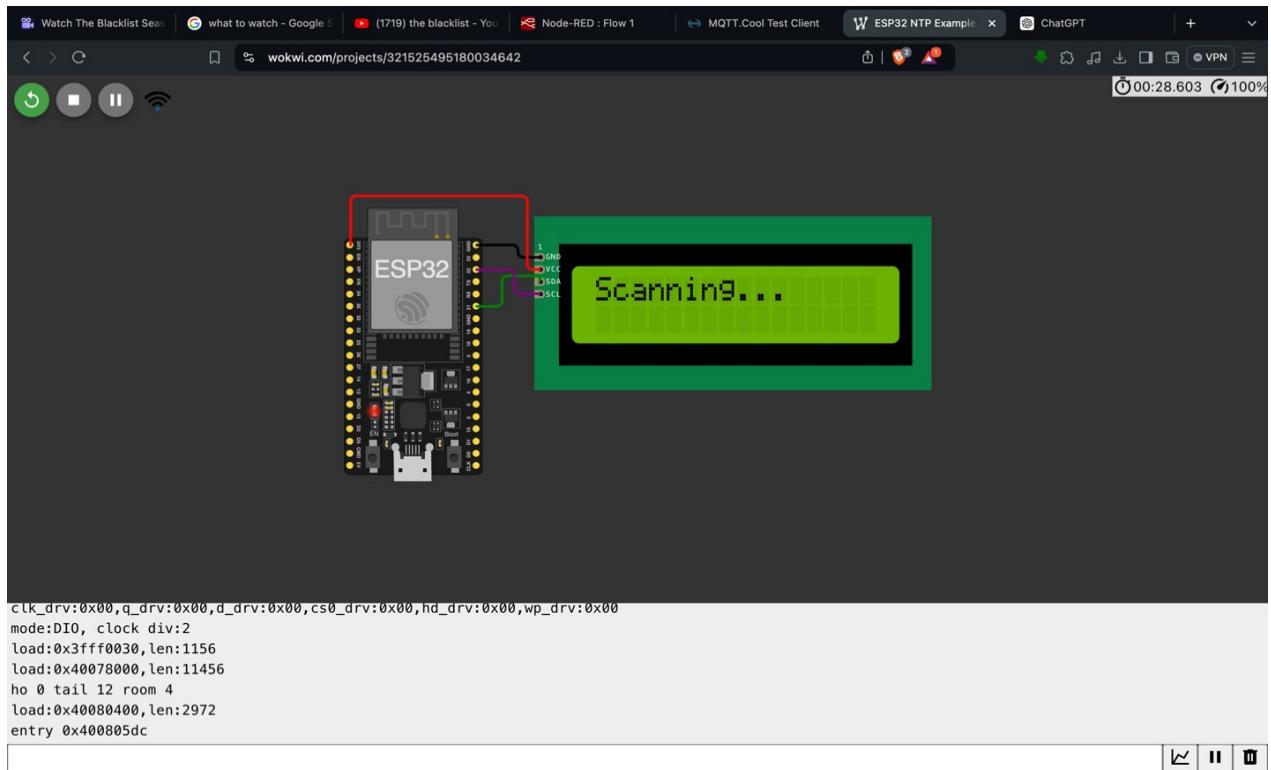


2nd question:

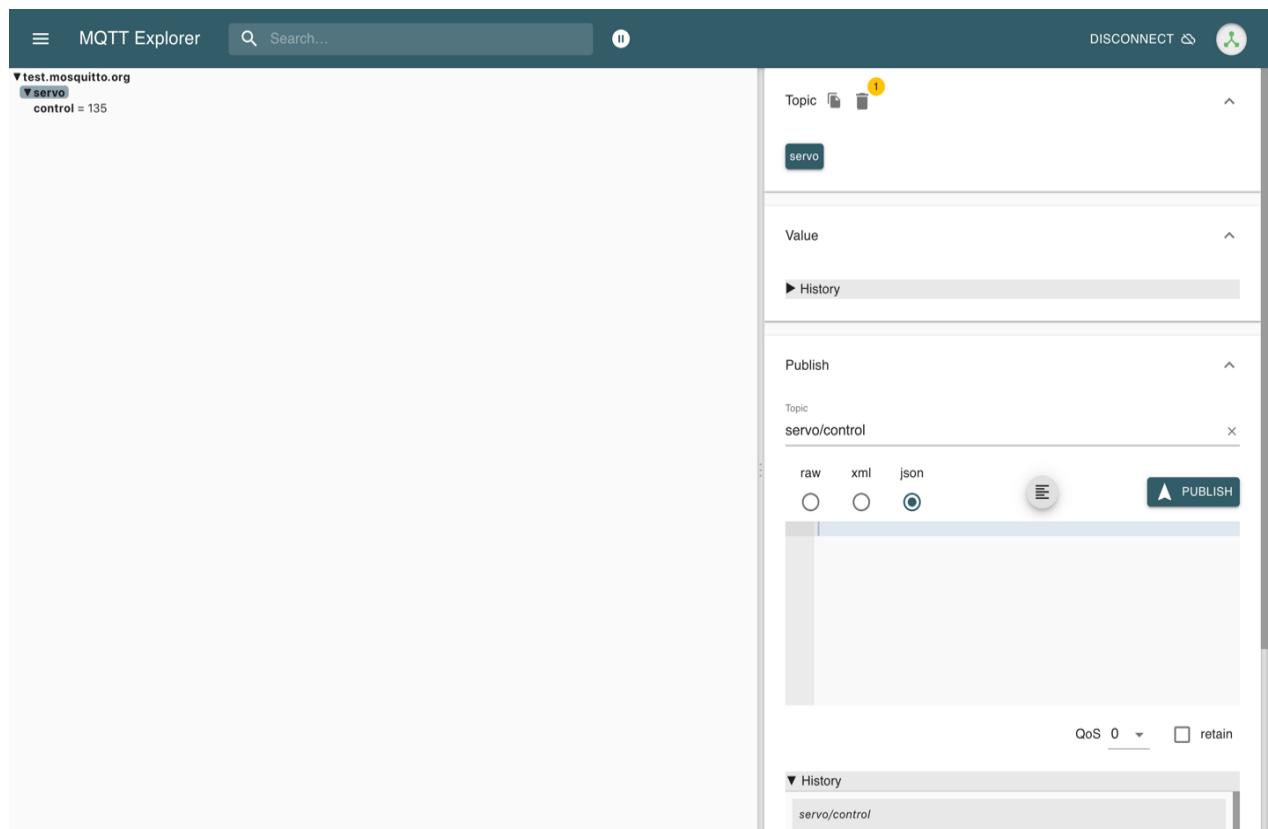
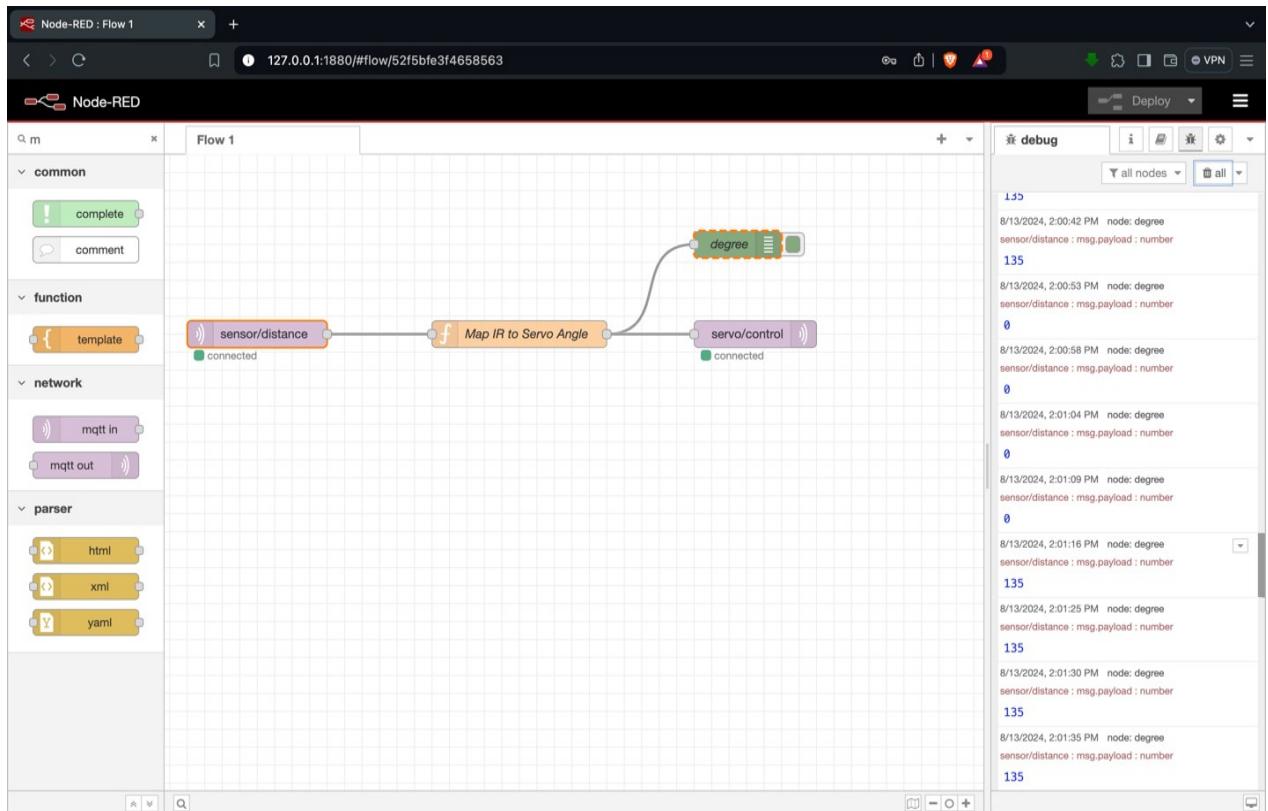


```
1 #include <WiFi.h>
2 #include <ESPAsyncWebServer.h>
3
4 // Replace with your network credentials
5 const char* ssid = "WE_9BFF74";
6 const char* password = "*****";
7
8 // GPIO pins where the LEDs are connected
9 const int ledPin1 = 27;
10 const int ledPin2 = 26;
11
12 // Create AsyncWebServer object on port 80
13 AsyncWebServer server(80);
14
15 void setup() {
16     // Initialize Serial Monitor
17     Serial.begin(9600);
18
19     // Set LED pins as outputs
20     pinMode(ledPin1, OUTPUT);
21     pinMode(ledPin2, OUTPUT);
22
23     // Ensure LEDs are off initially
24     digitalWrite(ledPin1, LOW);
25     digitalWrite(ledPin2, LOW);
26
27     // Connect to Wi-Fi
28     Serial.print("Connecting to ");
29     Serial.println(ssid);
30     WiFi.begin(ssid, password);
31     while (WiFi.status() != WL_CONNECTED) {
32         delay(1000);
33         Serial.println("Connecting to WiFi...");
34     }
35     Serial.println("Connected to WiFi");
36
37     // Print the IP address
38     Serial.print("IP Address: ");
39     Serial.println(WiFi.localIP());
40
```

3rd question:



4th question



Screenshot of the VS Code IDE showing the main.cpp file for an ESP32 project titled "IR_MQTT_SERVO".

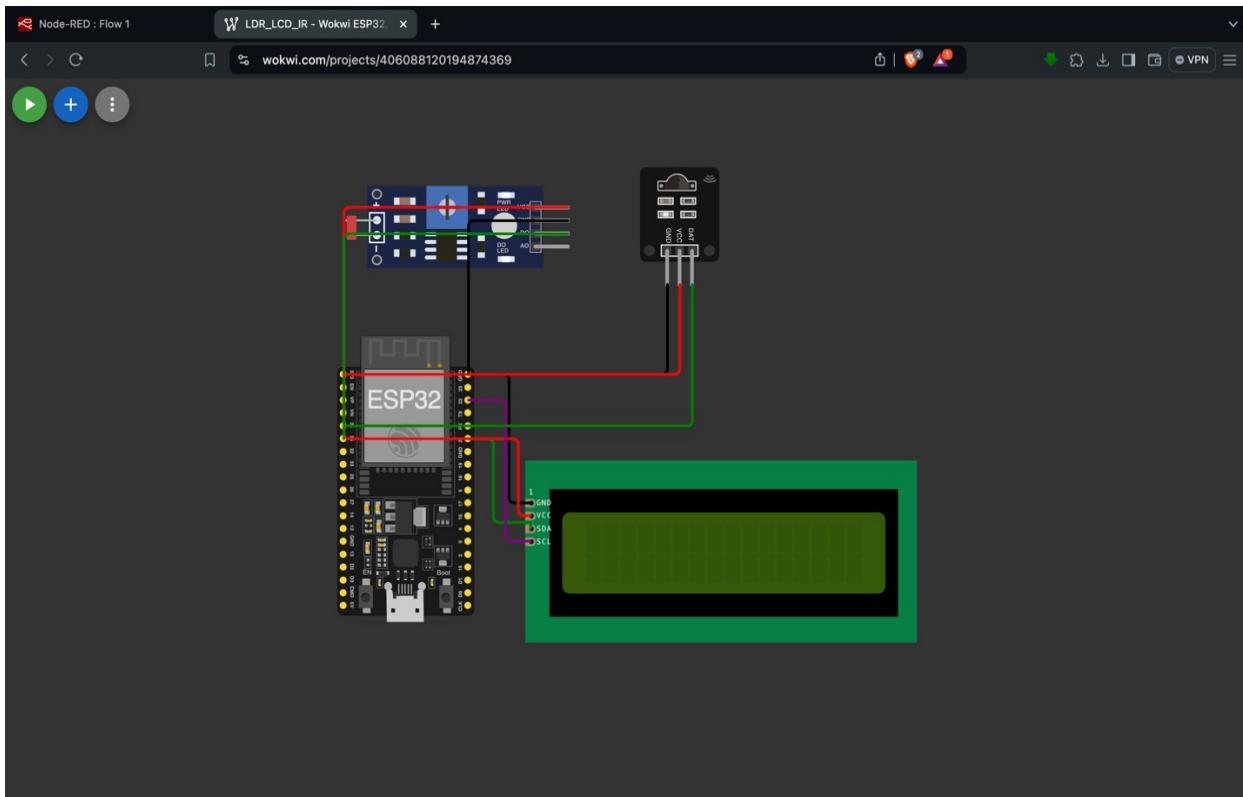
```

EXPLORER    ...   C\ main.cpp x
src > C\ main.cpp > loop()
60 void setup()
61 {
62     delay(1000);
63     Serial.println("Connecting to WiFi..");
64 }
65 Serial.println("Connected to WiFi");
66
67 // This bypasses certificate verification (not recommended for production)
68 espClient.setInsecure();
69
70 client.setServer(mqtt_server, mqtt_port);
71 client.setCallback(callback);
72
73 if (client.connect("ESP32_Client"))
74 {
75     Serial.println("Connected to MQTT broker");
76     client.subscribe("servo/control");
77 }
78 else
79 {
80     Serial.print("Failed to connect to MQTT broker, state: ");
81     Serial.println(client.state());
82 }
83
84
85
86
87
88
89
90 }
91
92 void loop()
93 {
94     if (!client.connected())
95     {
96         reconnect();
97     }
98     client.loop();
99
100    int irValue = analogRead(irSensorPin);
101    Serial.print("IR Sensor Value: ");
102    Serial.println(irValue);
103
104    // Publish IR sensor value to MQTT topic
105    client.publish("sensor/distance", String(irValue).c_str());
106
107    delay(2000); // Adjust the delay as needed
108
109
110
111

```

The code implements a basic MQTT client for an ESP32. It connects to a WiFi network and subscribes to the "servo/control" topic. It also publishes the value from an IR sensor connected to pin D1 to the "sensor/distance" topic every 2 seconds.

5th question:



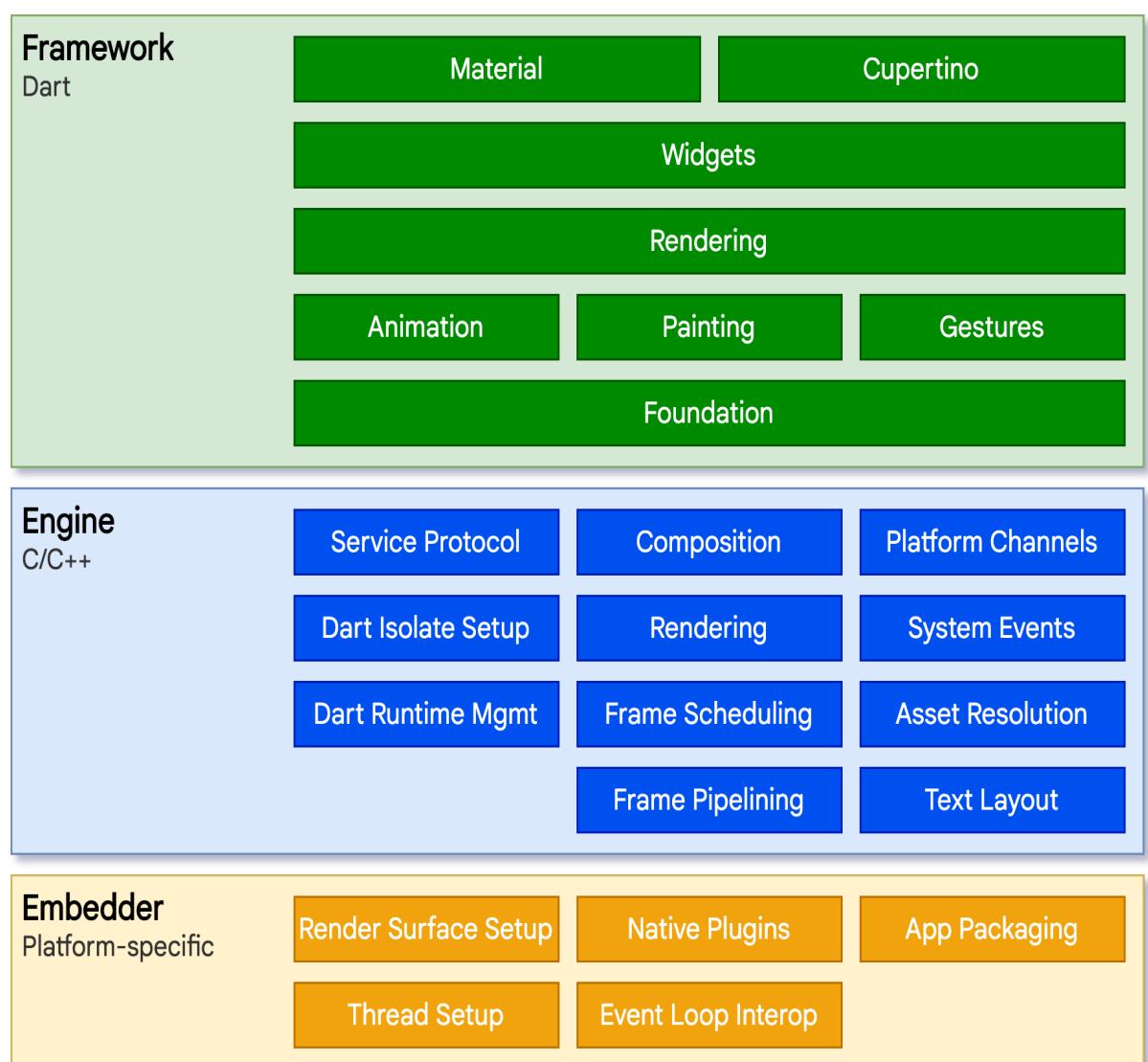
```
1 void callback(char* topic, byte * payload,
2               unsigned int length) {
3     Serial.print("Message arrived [");
4     Serial.print(topic);
5     Serial.print("] ");
6     String message;
7     for int i = 0; i < length; i++) {
8         message += payload[i];
9     }
10    Serial.println(message);
11 }
12
13 void setup() {
14     Serial.begin(9600);
15     setup_wifi();
16     client.setServer(mqtt_server, 1883);
17     client.setCallback(callback);
18 }
19
20 void loop() {
21     if (!client.connected()) {
22         reconnect();
23     }
24     client.loop();
25
26     int ldrValue = analogRead(LDR_PIN);
27     int irValue = analogRead(IR_PIN);
28
29     String payload = "{\"ldr\":\"";
30     payload += ldrValue;
31     payload += "\", \"ir\":\"";
32     payload += irValue;
33     payload += "}";
34
35     Serial.print("Publishing message: ");
36     Serial.println(payload);
37
38     client.publish("sensor/data", (char *)
39                    payload.c_str());
40     delay(2000);
41 }
42
```



- x. I've worked on any of the theoretical questions in the assignment but I've done my research about each of the topics included all over.

3. Application Design and Deployment using Flutter

- i. I had no background of flutter, dart, firebase what so ever so I had to learn all of them from the beginning so it took me quite some time specially that flutter's syntax and style is so much different than any programming language I've used before and closer to java script than it is to java or python so it took me some time to learn and understand how it works so somewhere between 10 to 25 hours.
- ii. Eng/ Mahmoud Essam.
- iii. The outcome was for me to learn the programming language itself and to be able to create and develop programs in it and also learn more about the firebase what is it and how it's used and why exactly.
- iv. The theoretical parts were dart and flutter's syntax and why flutter not swift or Kotlin and what is the firebase and what it's used for and how.
- v.





Build better apps



Improve app quality



Grow your app



Auth



Hosting



Crashlytics



Analytics



Remote Config



Cloud Functions



ML Kit



Performance Monitoring



Predictions



A/B Testing



Cloud Firestore



Realtime Database



Test Lab



Cloud Messaging



Dynamic Links



Cloud Storage



In-app Messaging

The screenshot shows the Firebase Analytics dashboard for the 'AstroPals' project. The top navigation bar includes links for Home, Analytics, Develop, Grow, Notifications, Dynamic Links, EARN, AdMob, Spark Free, and Upgrade.

The main dashboard features several key metrics:

- Active users:** A line chart showing user engagement over time (Apr - May). It includes three lines representing different segments: Active users (~2.5M), New users (~1.3M), and Returning users (~1.1M).
- Average revenue:** A table comparing ARPU (Average Revenue Per User) across different time periods:

| | Monthly | Weekly | Daily |
|-------|---------|---------|---------|
| ARPU | \$2.48 | \$1.76 | \$0.31 |
| ARPPU | \$25.05 | \$23.13 | \$42.15 |
- Attribution:** A table showing first_open attribution for 160 days ending May 11:

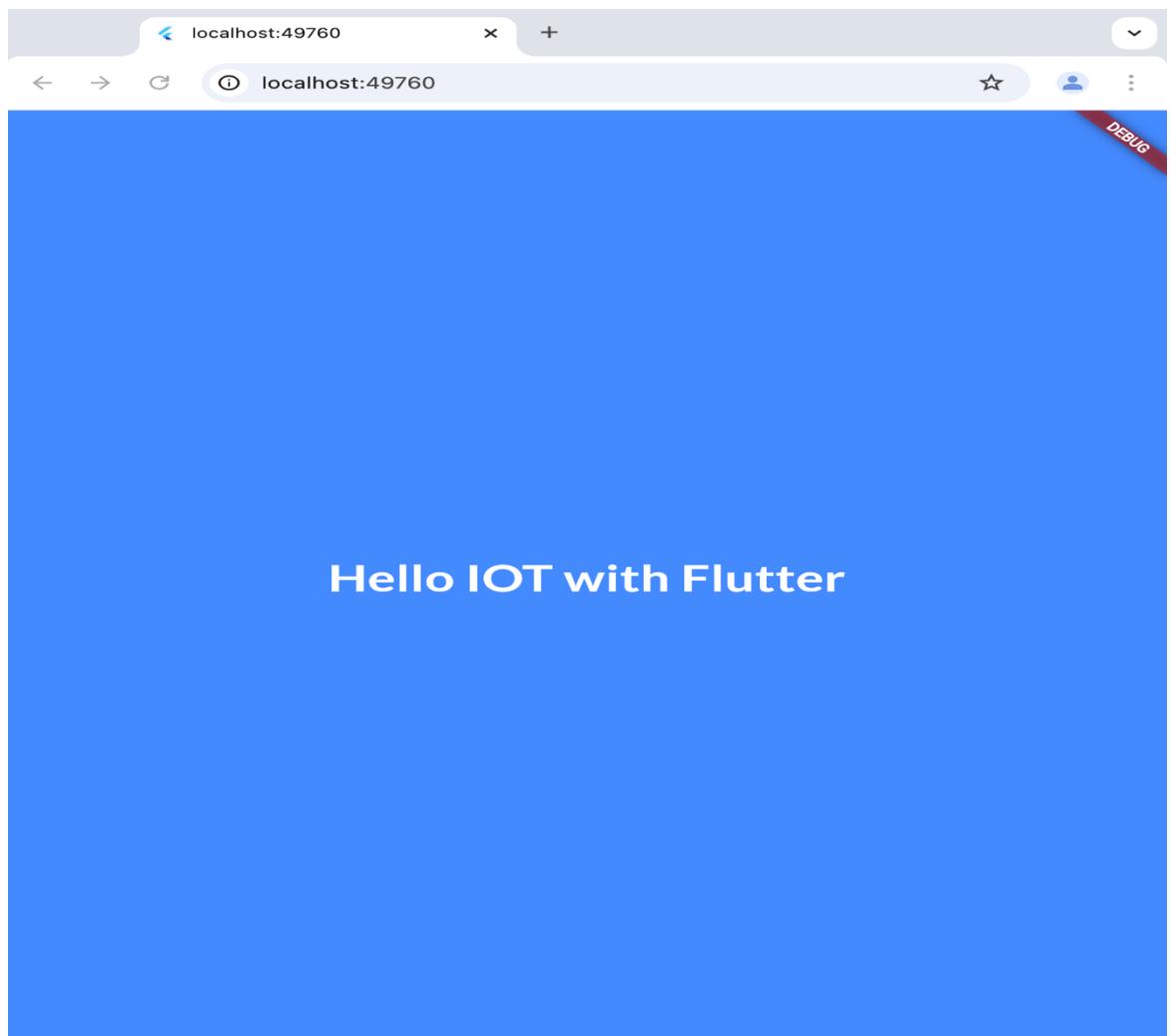
| Source | First Open | Lifetime Value (LTV) |
|---------------|------------|----------------------|
| (Direct) | 15K | \$400.55 |
| Google | 38K | \$265.32 |
| Adrise | 24K | \$15.50 |
| AlphaApps | 12K | \$10.20 |
| Alte Ads | 11K | \$8.50 |
| Stomp Network | 6K | \$7.90 |
- Retention cohorts:** A heatmap showing retention rates for various cohorts over 6 weeks ending May 11:

| | Week 0 | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
|-----------------|--------|--------|--------|--------|--------|--------|
| Mar 27 - Apr 2 | 100% | 19.5% | 12.3% | 10.4% | 8.4% | 8% |
| Apr 3 - Apr 9 | | | | | | |
| Apr 10 - Apr 16 | | | | | | |
| Apr 17 - Apr 23 | | | | | | |
| Apr 24 - Apr 30 | | | | | | |
| May 1 - May 7 | | | | | | |

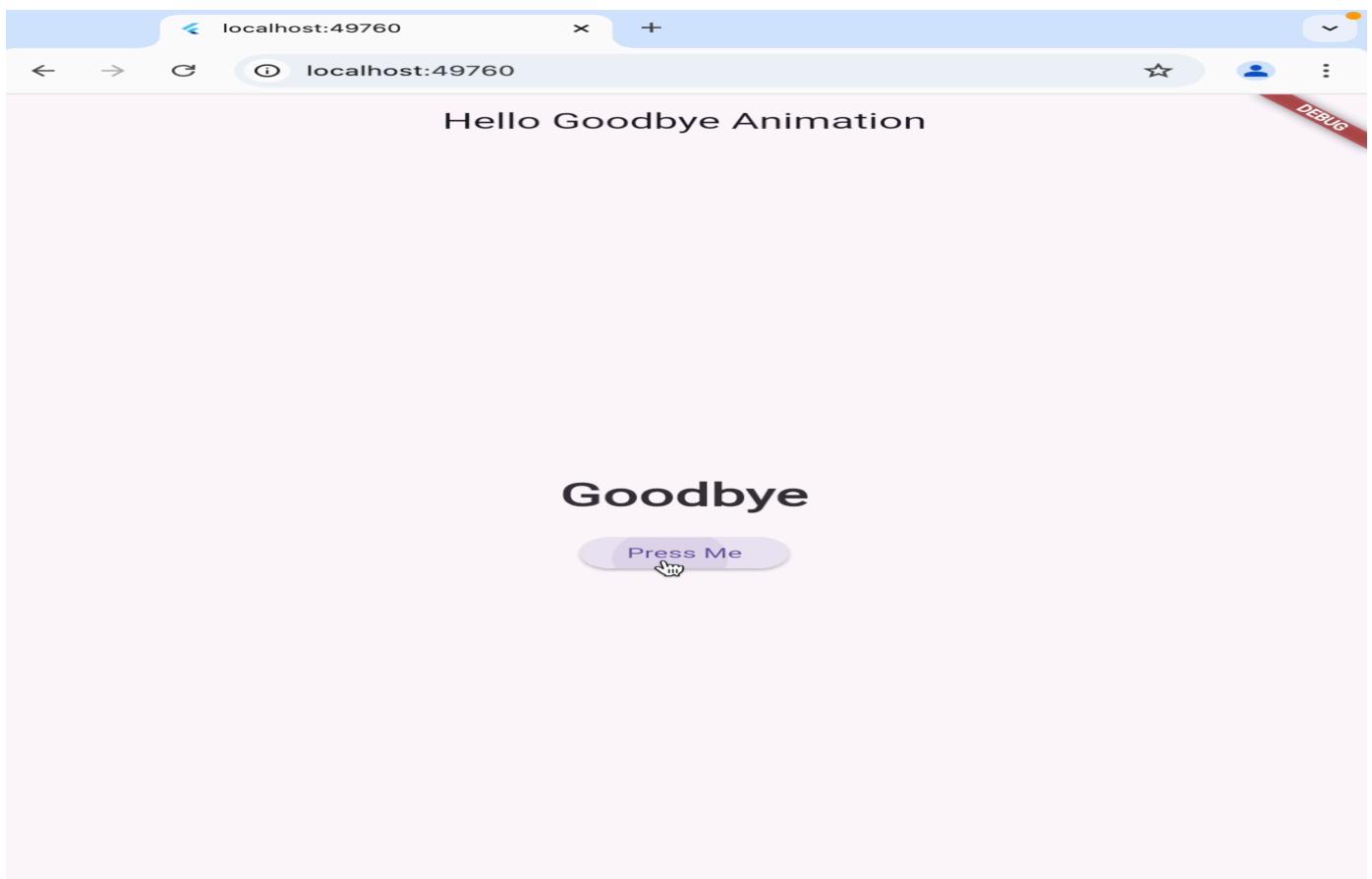
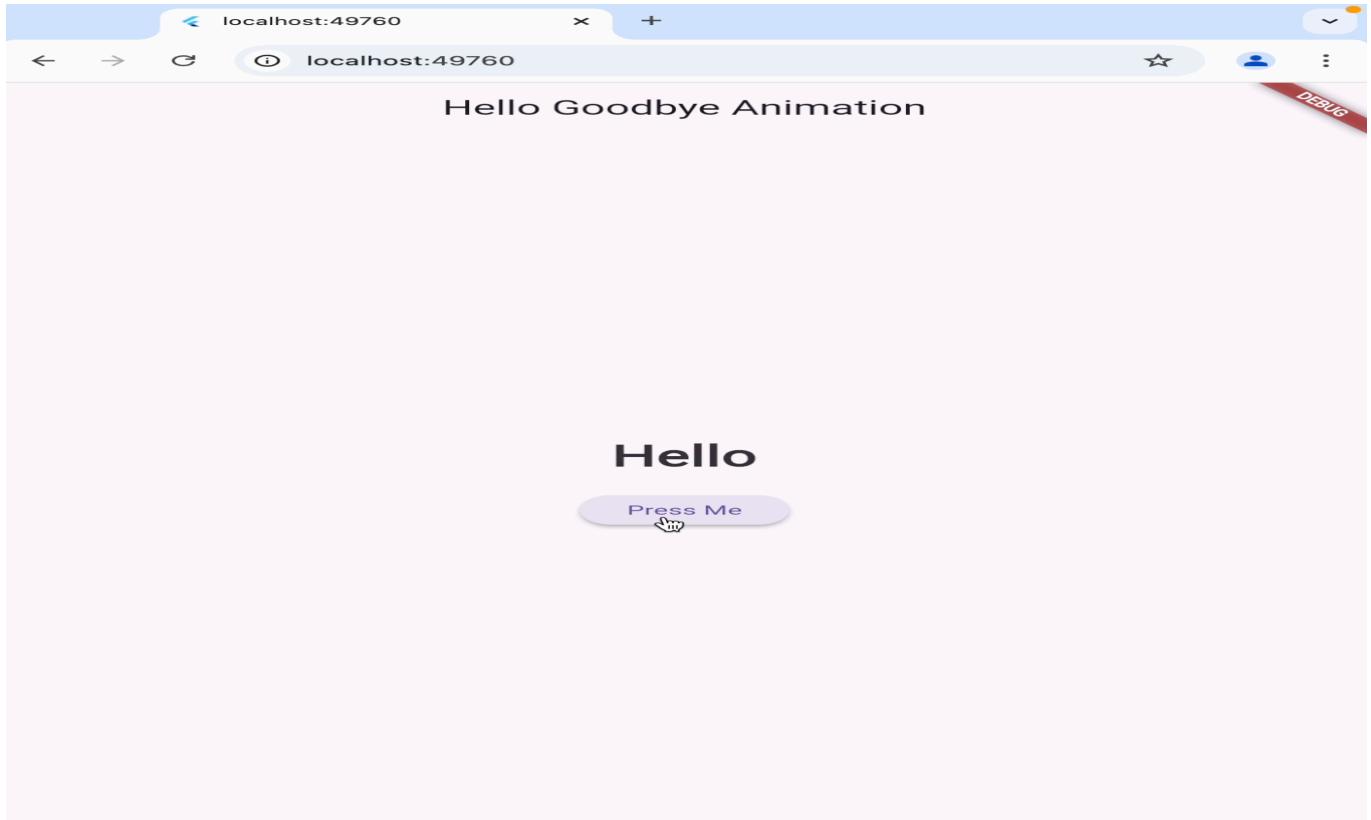
- vi. The onsite practice were 2 questions one of them was to write down my team's names and mine in a container with a background of a different

colour to that container than the rest of the programme and the other one was to create two pages with a homepage where you get to navigate to and through all those pages to each other smoothly.

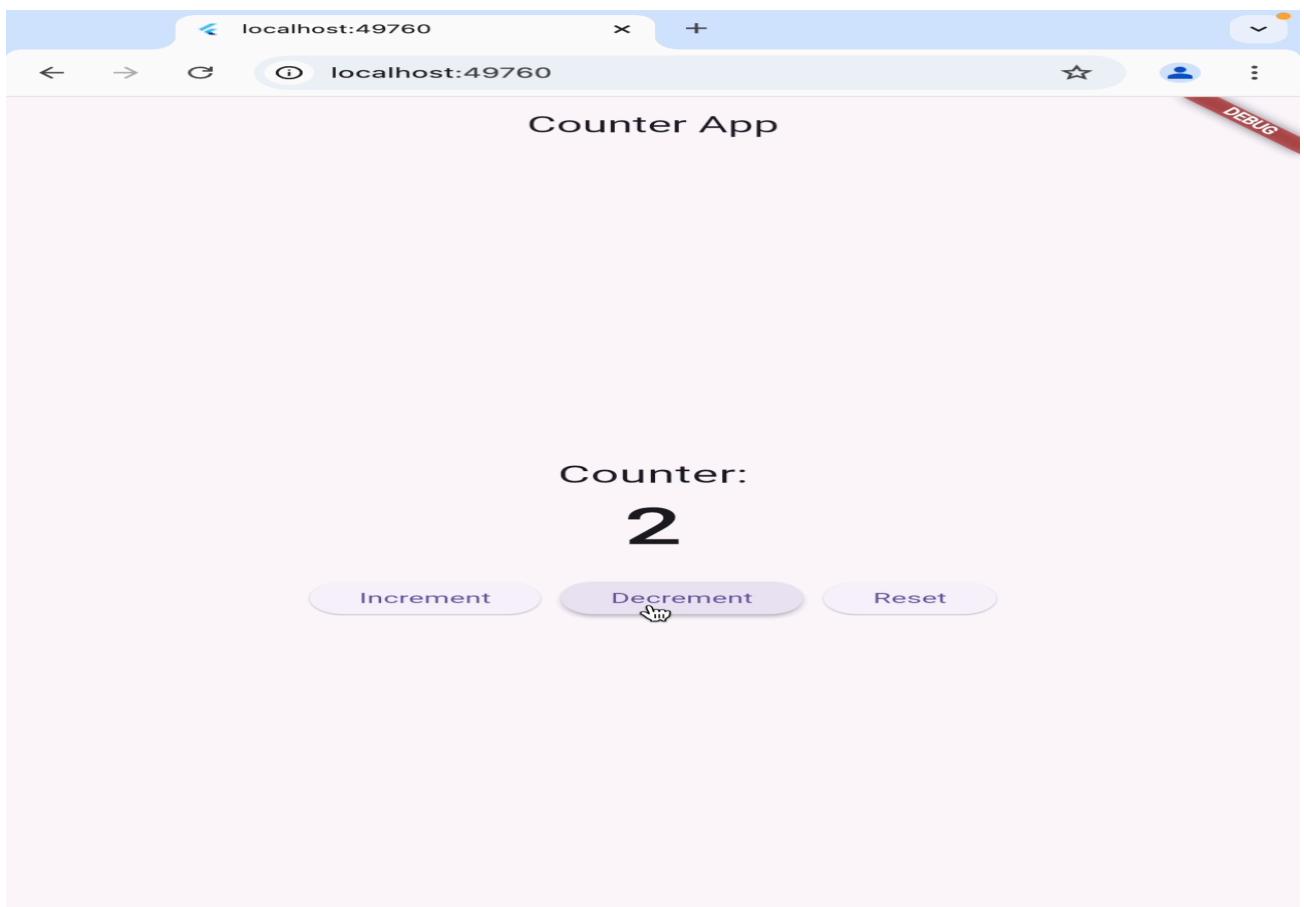
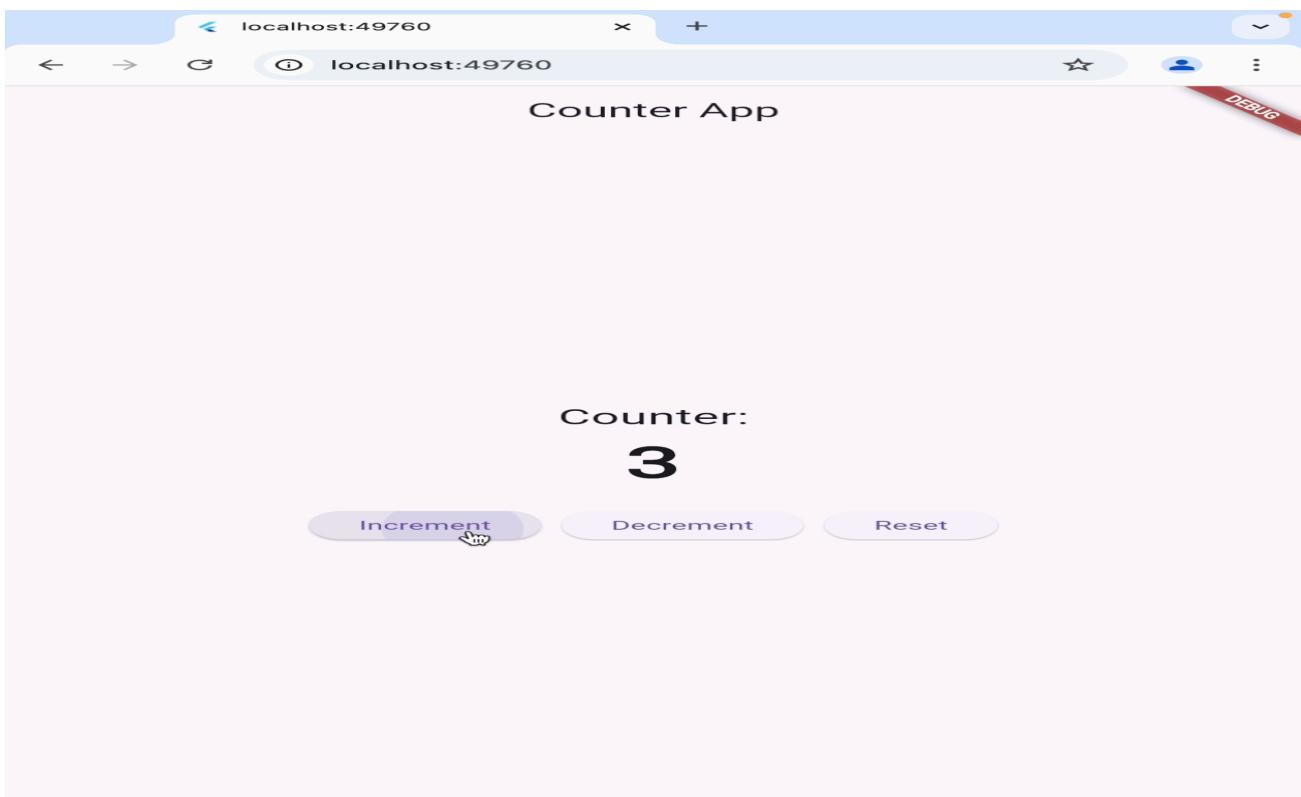
- vii. I don't have the screens of them right now because I've deleted them to free up some space for the rest of the assignment programs but there're similar questions in the assignment itself and I worked on them and will be included in the next point.
- viii. I worked on the first theoretical question and 2,3,4,5 and because I was also working on the final project the idea setup and everything I didn't give that assignment that much care but I did really work as hard as I could in the projects app and added a lot of features and made something special not just a copy of any other app so I'd show that I did work on the last chapter as hard as I did on the first three and the drive included in that module pdf that I handed in was not completely uploaded and I didn't really notice till now so I'll upload all the codes and add screens of them also and they'll be included in the QR of that chapter.
- ix. 2nd question:

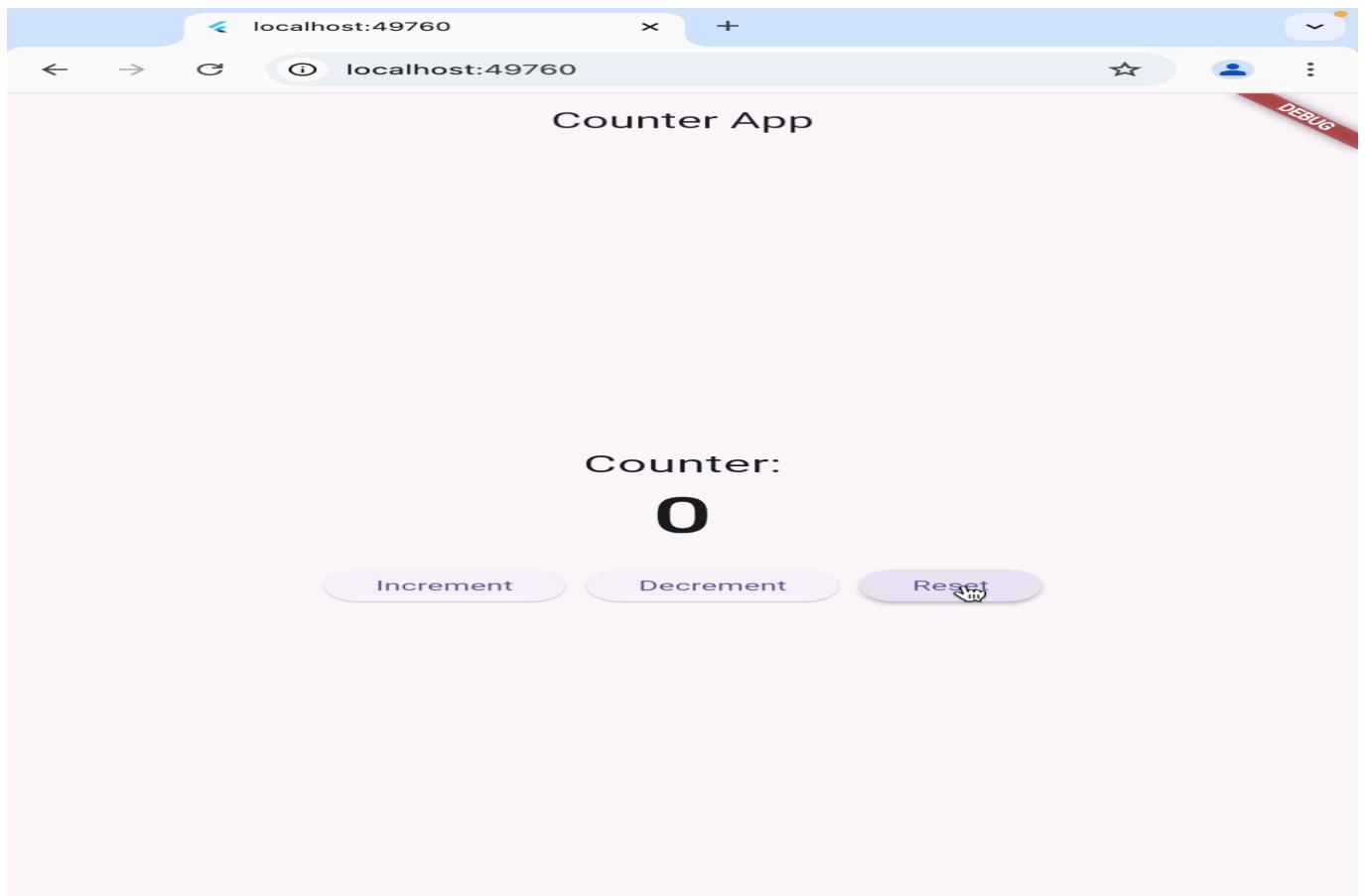


3rd question:

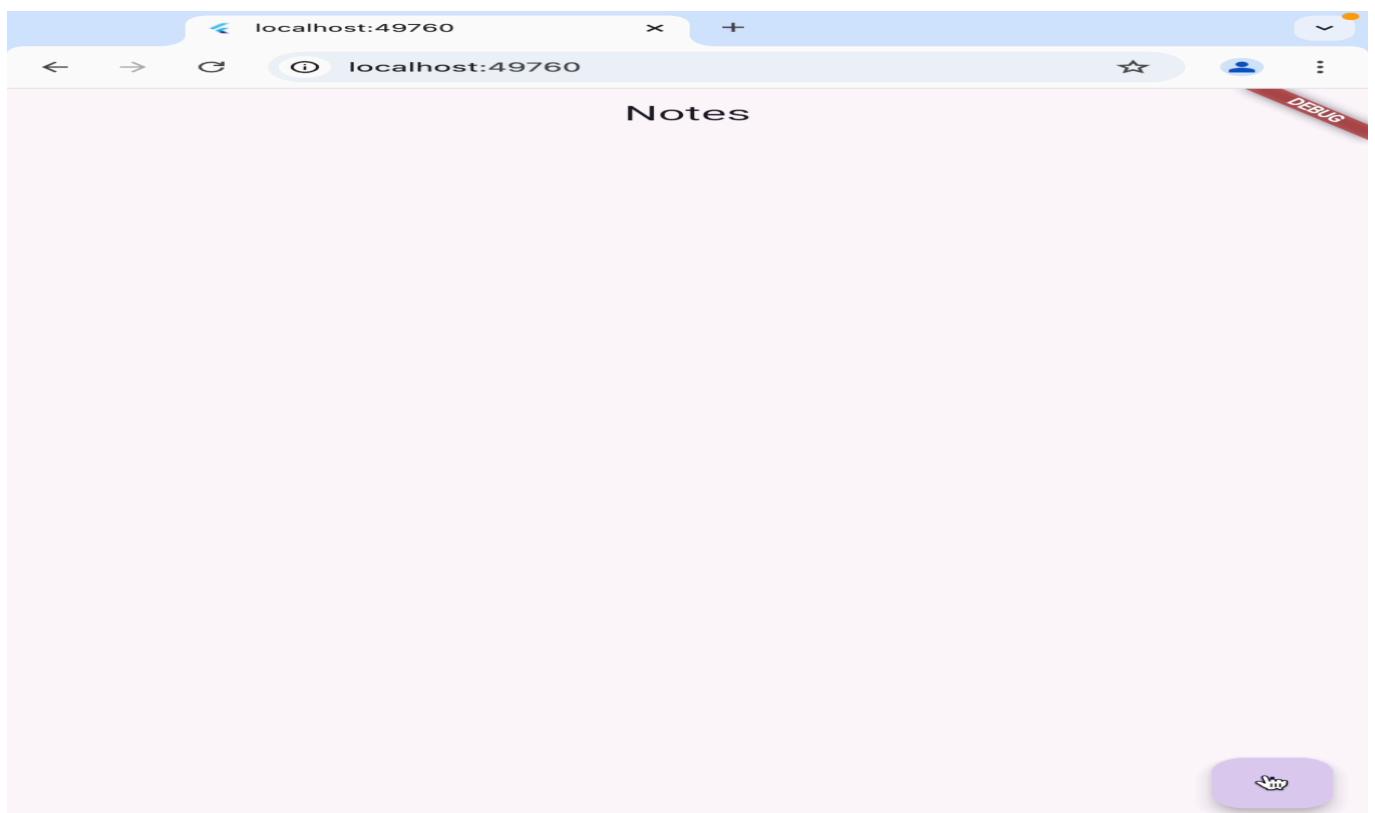


4th question:



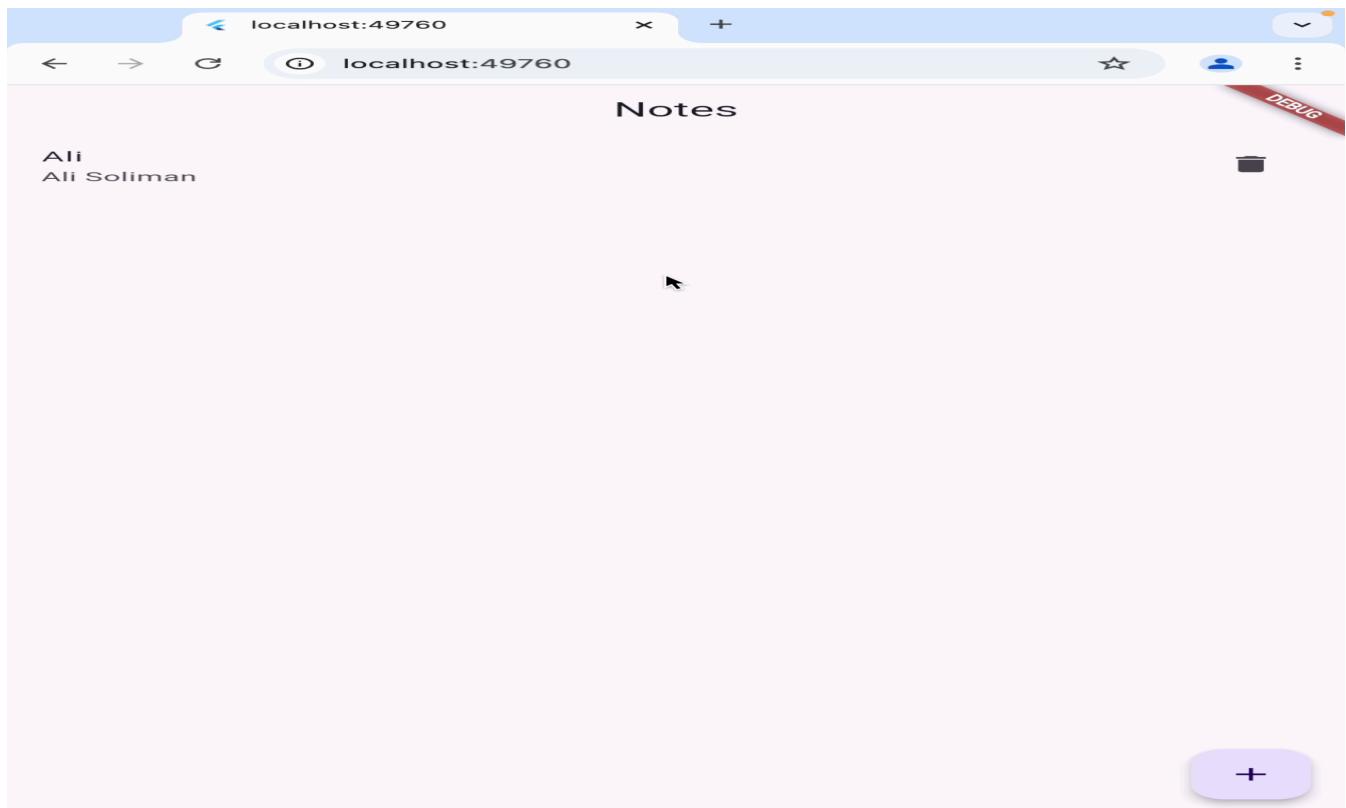


5th question:



A screenshot of a mobile browser window titled "localhost:49760". The main content is an "Add Note" form. It features two input fields: "Title" and "Description", both currently empty. Below the fields is a blue "Add Note" button.

A screenshot of a mobile browser window titled "localhost:49760". The main content is an "Add Note" form. The "Title" field contains the text "Ali". The "Description" field contains the text "Ali Soliman". A cursor icon is positioned over the blue "Add Note" button.



All the links videos and everything else:



- x. I've worked also on the first theoretical question.

4. Ai in IoT

- i. 5 to 10 hours.
- ii. Eng/ Fares Hazem.
- iii.
 1. Enhanced Data Analysis and Decision-Making:
 - Predictive Maintenance: AI algorithms can analyze data from IoT sensors to predict when equipment is likely to fail, enabling proactive maintenance and reducing downtime.
 - Real-Time Insights: AI can process large volumes of data from IoT devices in real-time, providing actionable insights and enabling quicker decision-making.
 2. Automation and Efficiency:
 - Smart Home Automation: AI-powered IoT devices can automate household tasks, such as adjusting thermostats, lighting, and security systems, based on user preferences and behaviors.
 - Industrial Automation: In manufacturing, AI can optimize production processes, monitor equipment health, and manage supply chains more efficiently.
 3. Improved Security and Surveillance
 - Anomaly Detection: AI can detect unusual patterns in data from IoT devices, identifying potential security threats or operational anomalies.
 - Enhanced Surveillance: AI-enhanced cameras and sensors can provide better security through facial recognition, motion detection, and behavior analysis.
 4. Personalization and User Experience
 - Smart Assistants: AI in IoT enables devices like smart speakers and virtual assistants to learn from user interactions, providing more personalized and intuitive experiences.
 - Customized Services: AI can tailor services and products to individual users based on data collected from IoT devices, enhancing customer satisfaction.
 5. Energy Management and Sustainability
 - Energy Optimization: AI algorithms can analyze data from IoT devices to optimize energy usage in buildings, reducing costs and environmental impact.

- Sustainable Practices: IoT devices, combined with AI, can monitor and manage resources more efficiently, promoting sustainable practices in various industries.

6. Healthcare Advancements

- Remote Monitoring: AI-powered IoT devices can monitor patients' health remotely, analyzing vital signs and alerting healthcare providers to any issues.
- Personalized Medicine: AI can analyze data from wearable IoT devices to provide personalized health recommendations and treatments.

7. Smart Cities and Infrastructure

- Traffic Management: AI can analyze data from IoT sensors in traffic systems to optimize traffic flow and reduce congestion.
- Infrastructure Monitoring: AI can predict maintenance needs for infrastructure such as bridges and roads by analyzing data from IoT sensors.

8. Agricultural Efficiency

- Precision Farming: AI can analyze data from IoT devices in agriculture to optimize irrigation, monitor soil health, and predict crop yields, improving overall farm productivity.
- iv. That session was one of the bonus sessions so I didn't really have the time to attend the offline sessions or work on those assignments as I was so busy trying to create and fix my project but I've seen the recorded session not all of them but I'll finish them and work on the assignment because I want to maybe before the discussion maybe after but it'll be done.

f. IoT Security

- i. 5 to 10 hours.
- ii. Eng/ Fares Hazem.
- iii.

a. Positive Outcomes

1. Protection of Sensitive Data
 - Data Privacy: Ensuring that personal and sensitive data transmitted by IoT devices is encrypted and protected from unauthorized access.
 - Compliance: Meeting regulatory requirements such as GDPR, HIPAA, and others by safeguarding user data.
2. Prevention of Cyber Attacks
 - Reduced Vulnerabilities: Implementing security best practices and regular updates to minimize vulnerabilities that could be exploited by cyber attackers.
 - Threat Detection: Utilizing advanced security measures such as anomaly detection and AI-based threat analysis to identify and mitigate threats in real-time.
3. Reliability and Trustworthiness
 - System Integrity: Ensuring that IoT systems operate as intended without interference from malicious actors, maintaining the integrity and reliability of services.
 - User Trust: Building and maintaining user trust by demonstrating a commitment to securing IoT devices and protecting user data.
4. Operational Continuity
 - Minimized Downtime: Preventing disruptions to services and operations caused by security breaches, thus maintaining business continuity.
 - Resilience: Enhancing the resilience of IoT systems against attacks, ensuring they can continue to function even under adverse conditions.
5. Economic Benefits
 - Cost Savings: Reducing the financial impact of data breaches and cyber attacks, including potential fines, legal costs, and loss of business.
 - Market Competitiveness: Gaining a competitive edge by offering secure IoT products, which can be a key differentiator in the market.

b. Negative Outcomes of Poor Security

1. Data Breaches

- Exposure of Sensitive Information: Unauthorized access to personal, financial, or proprietary information due to insecure IoT devices.

- Identity Theft: Increased risk of identity theft and fraud from compromised personal data.

2. Cyber Attacks

- DDoS Attacks: IoT devices can be hijacked to form botnets, leading to Distributed Denial of Service (DDoS) attacks on critical infrastructure.
- Malware Infections: Insecure IoT devices can be used as entry points for malware, affecting the entire network.

3. Operational Disruptions

- Service Interruptions: Attacks on IoT systems can cause significant disruptions to services, impacting businesses and users.
- Safety Risks: In critical sectors like healthcare and industrial control systems, insecure IoT devices can pose safety risks to individuals and operations.

4. Financial Losses

- Cost of Breaches: Financial costs associated with data breaches, including legal fees, fines, and compensation to affected users.
- Loss of Business: Potential loss of customers and business opportunities due to reputational damage from security incidents.

5. Regulatory Penalties

- Non-Compliance Fines: Failing to comply with data protection regulations can result in significant fines and legal repercussions.
- Increased Scrutiny: Businesses may face increased regulatory scrutiny and requirements if they fail to secure their IoT devices adequately.

- v. Also That session was one of the bonus sessions so I didn't really have the time to attend the offline sessions or work on those assignments as I was so busy trying to create and fix by project but I've seen the recorded session not all of them but I'll finish them and work on the assignment because I want to maybe before the discussion maybe after but it'll be done.

3. Chapter 3: Training Capstone Project

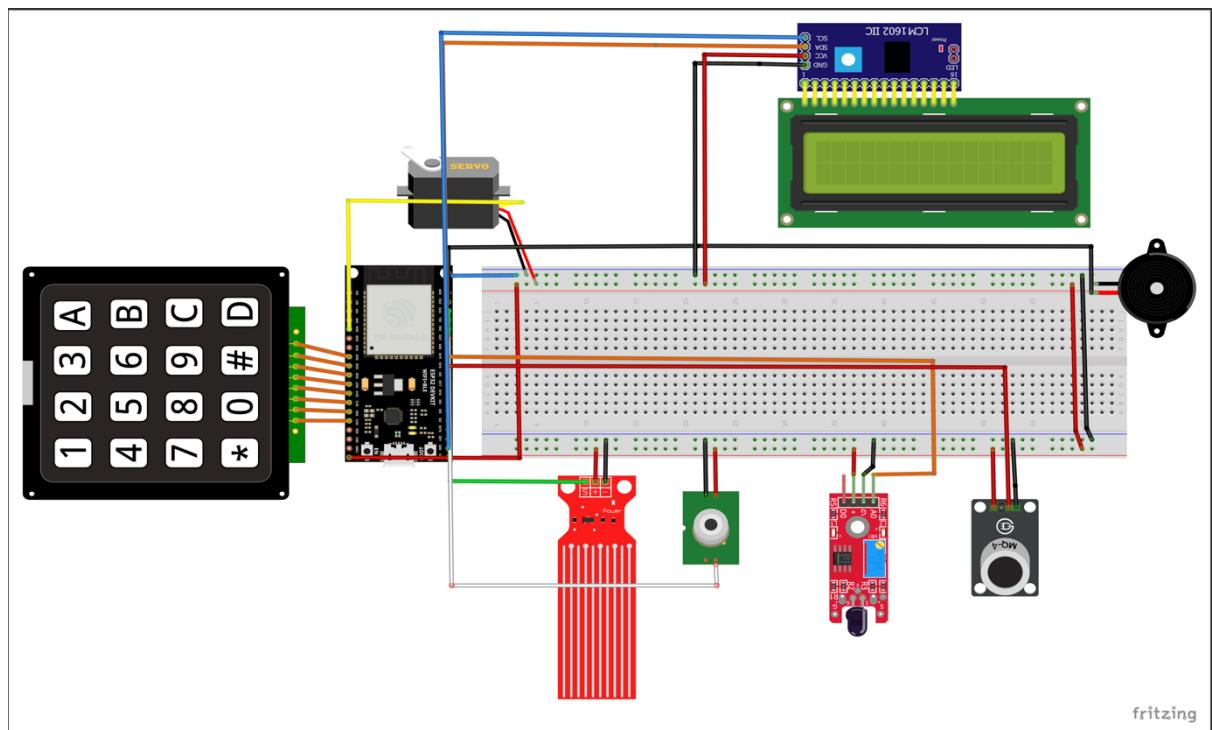
- i. Ali Mohamed Ali Soliman (22011564) Data Science
Youssef Ragab Sadeq (22010304) Data Science
Ahmed Hesham Ahmed Hafez (220111239) Data Science
Amr Mohamed Mahmoud Fayad (22010178) Data Science
Mohamed Ibrahim Fathy (22011956) Data Science
- ii. Ali Mohamed Ali Soliman: Hardware and design creation and implementation along with Flutter application with Firebase, MQTT and UIUX creation and deployment.

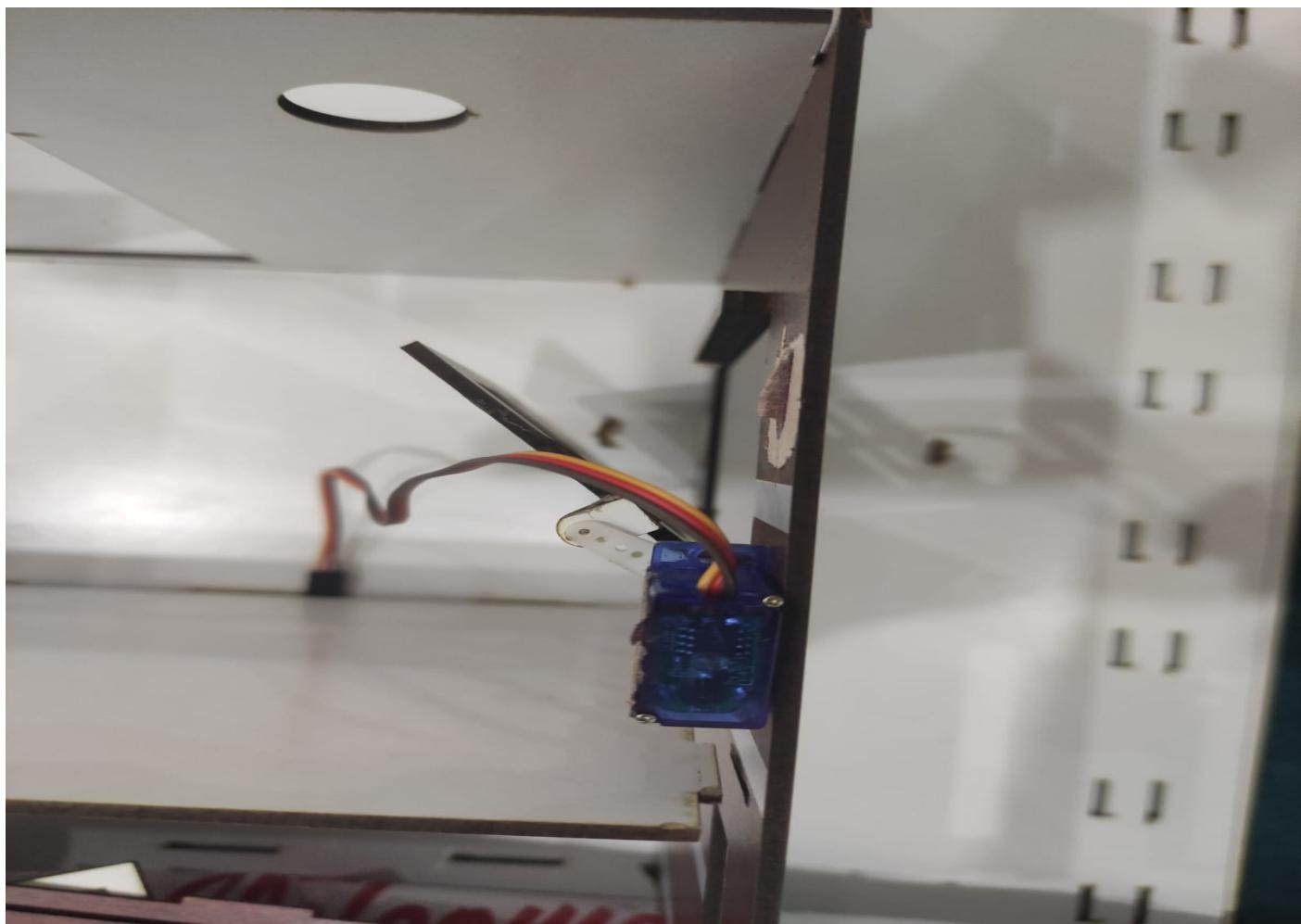
Youssef Ragab Sadeq: Hardware implementation along with Esp code with MQTT.

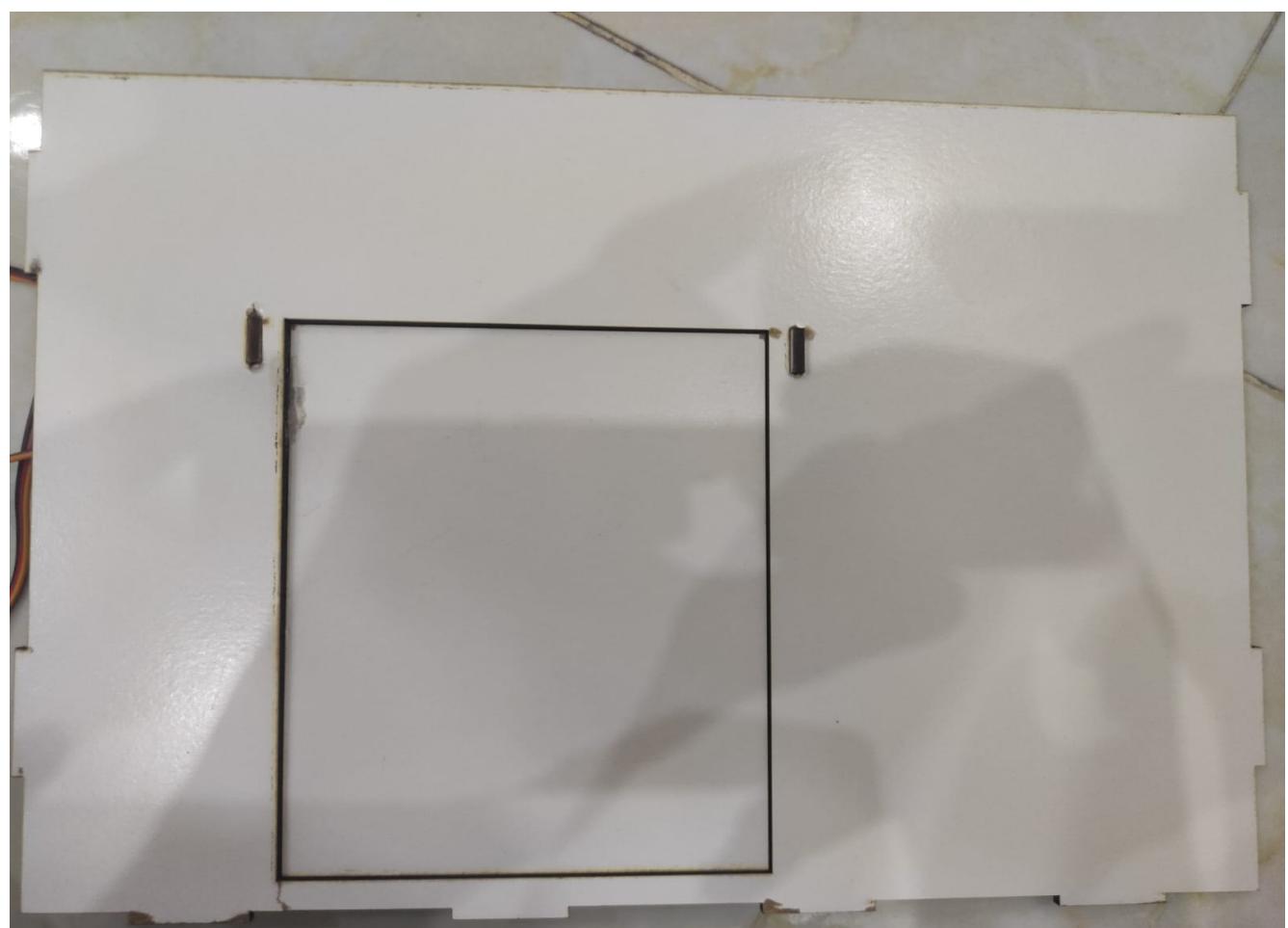
Ahmed Hesham Ahmed Hafez: LDR, RFID and buzzer implantation

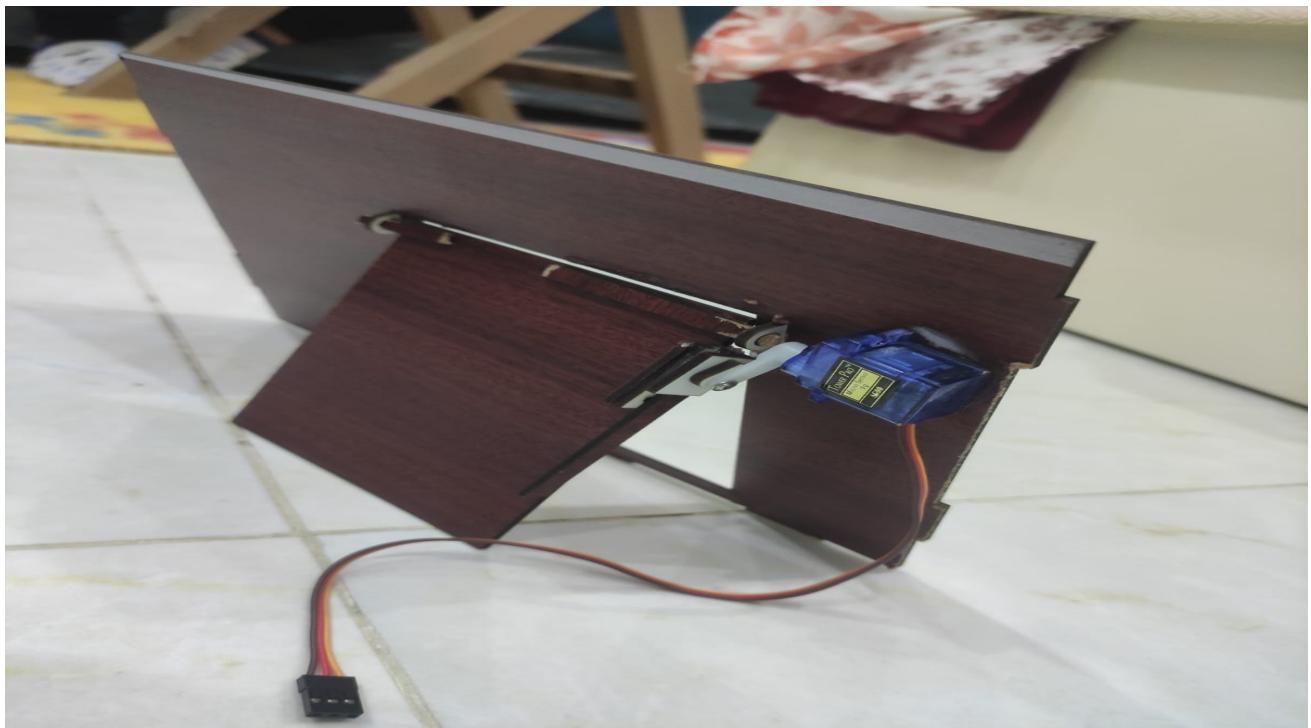
Amr Mohamed Mahmoud Fayad: full report & Implementing LEDs and Ir sensors

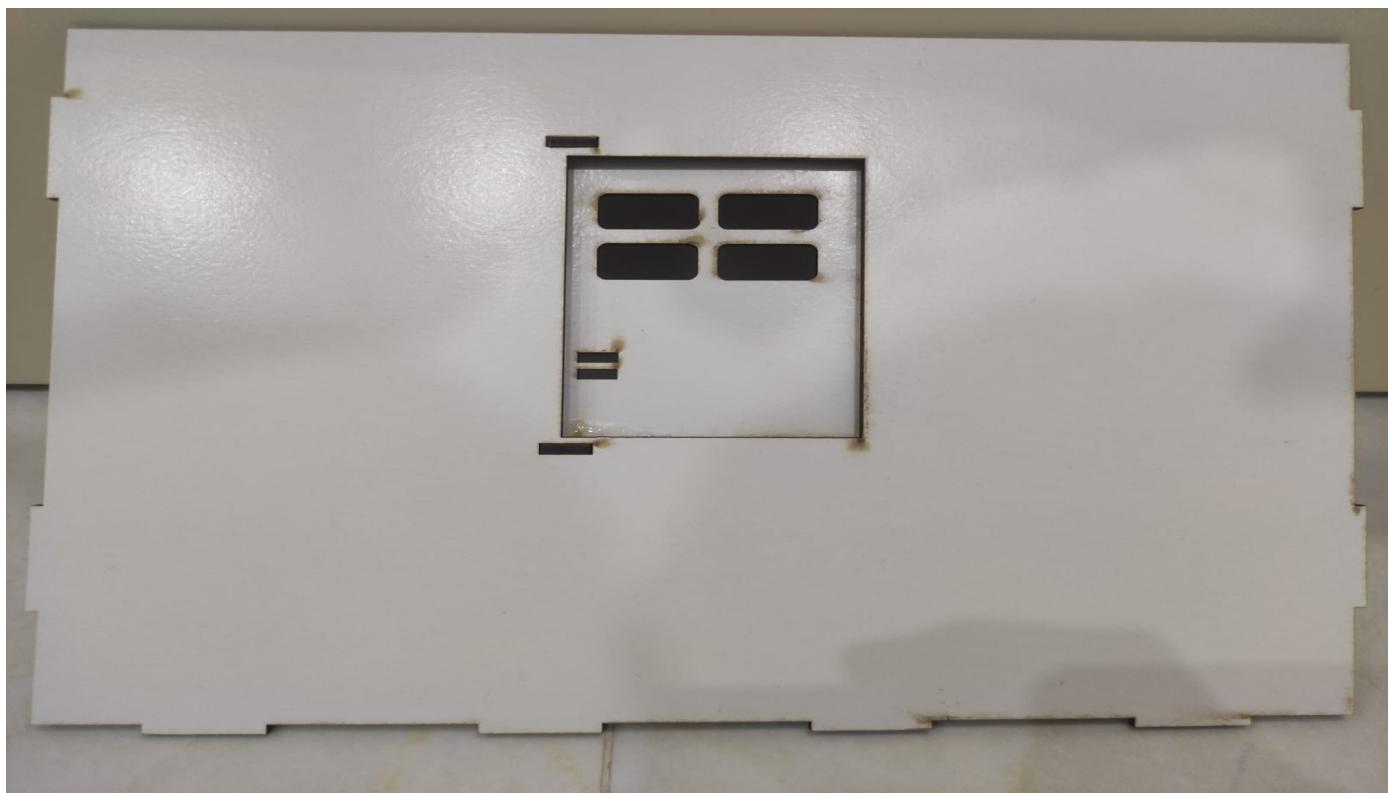
Mohamed Ibrahim Fathy: presentation & user interface and user ewperience.
- iii. The project required a lot of things more than 3 sensors, 1 or more actuator, real made maquette, schematic for the design, flutter application to control and monitor the sensors and motors.
- iv.



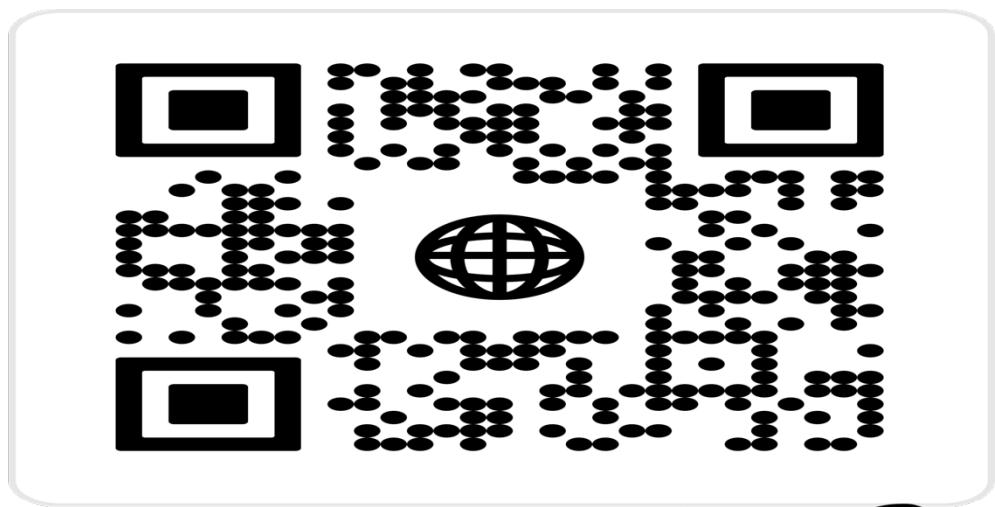
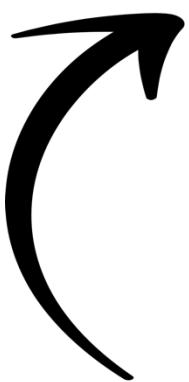






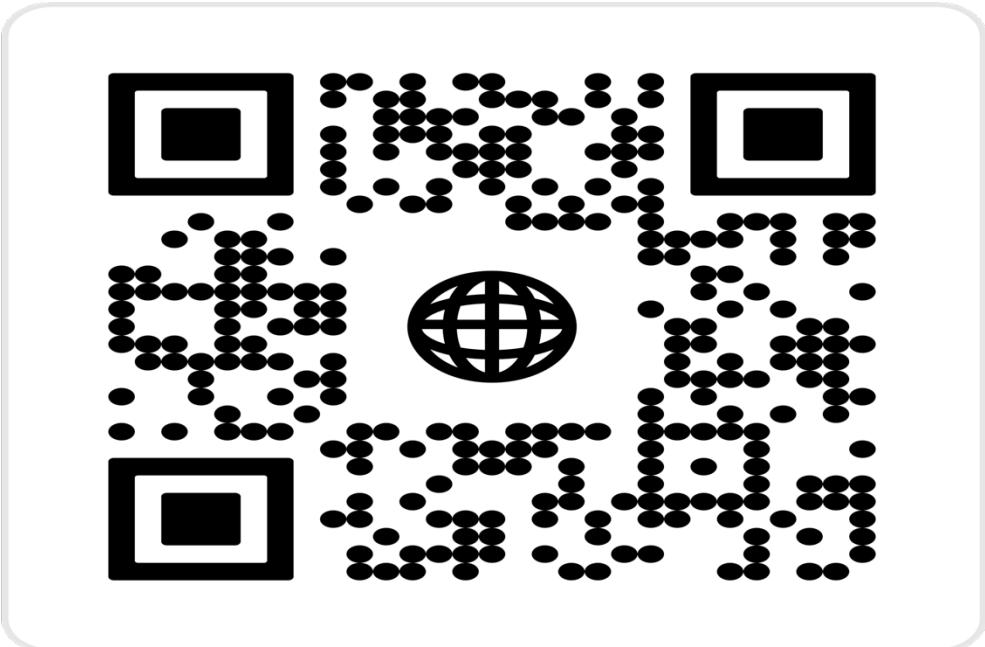
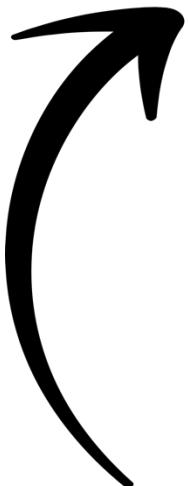


v. Videos



scan me

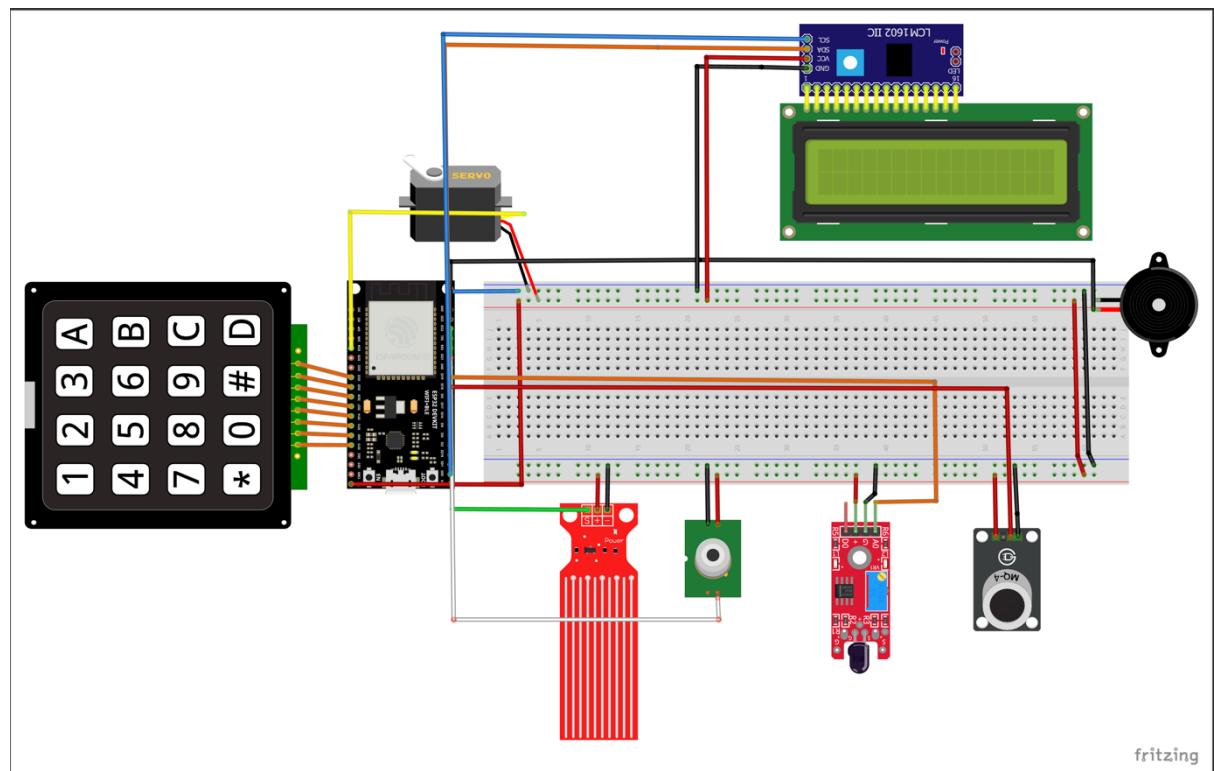
presentation



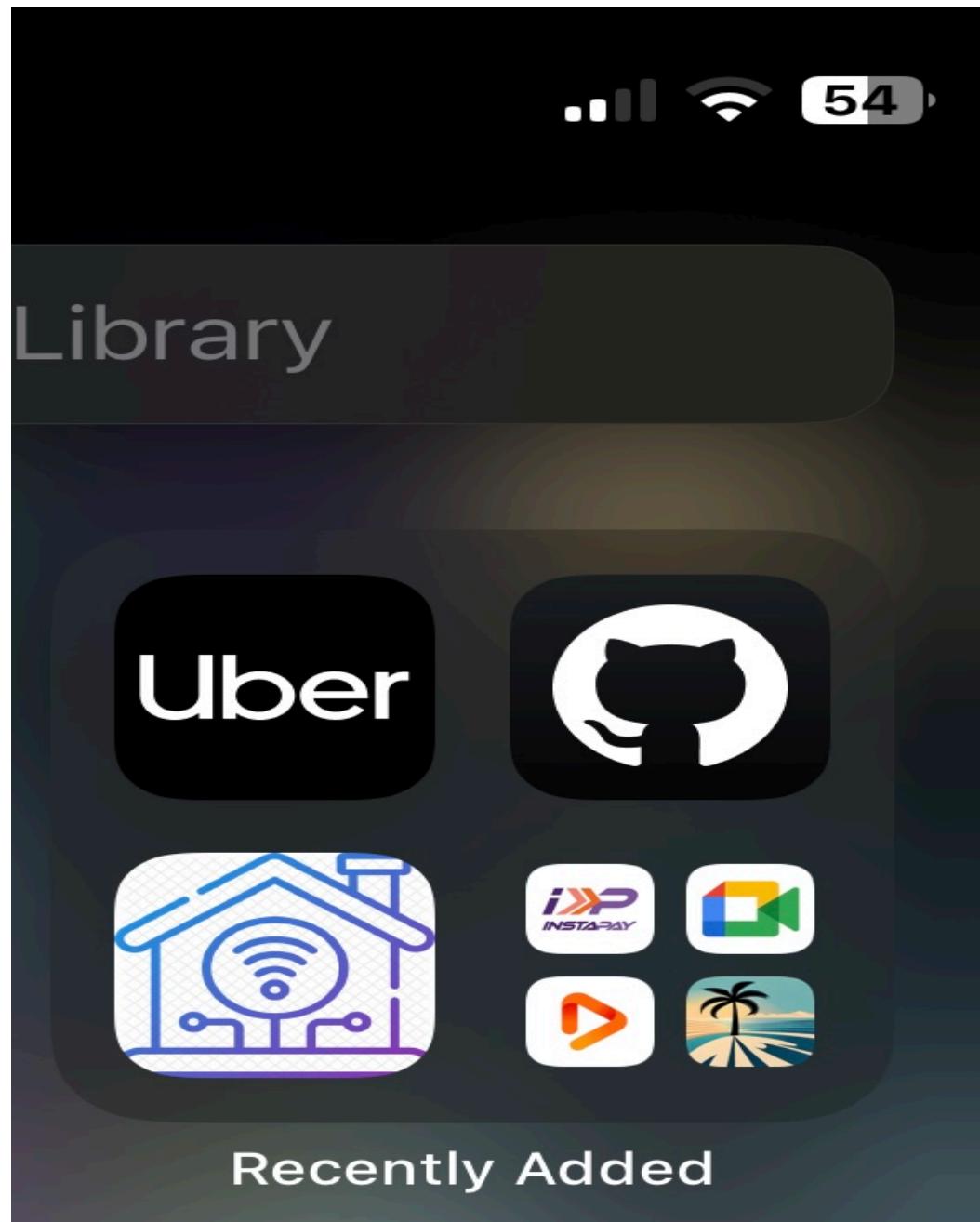
scan me

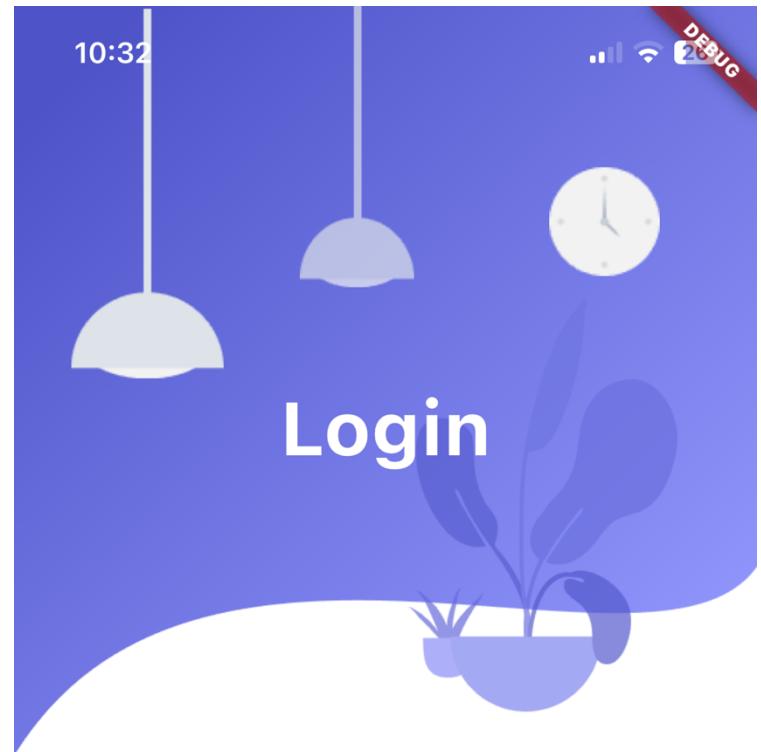
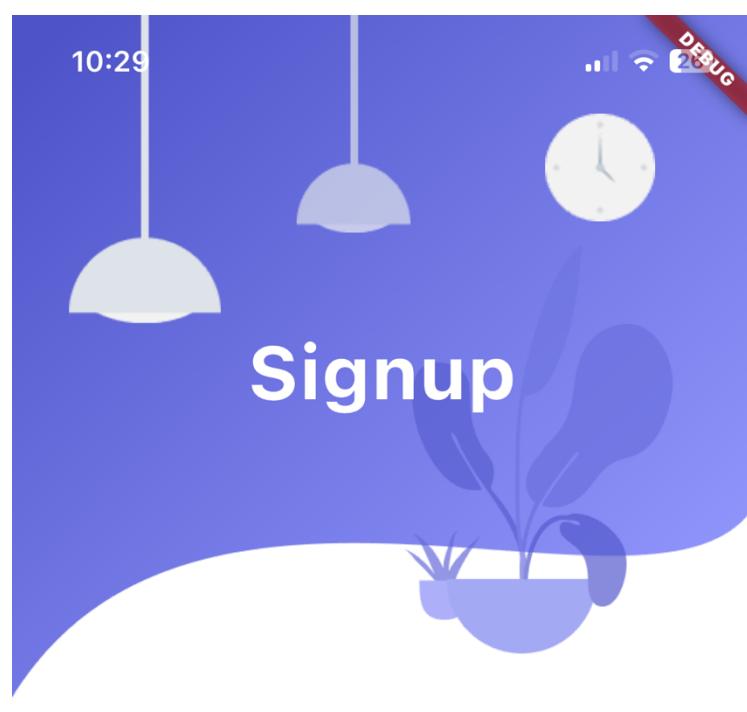
vi. I wanted to make something different and special not just anything to get the degree of the final project I wanted everything to be better than expected much better so I didn't want to make the maquette itself out of carton that could easily be ruined but made out of wood that's more premium and realistic and much more durable and expensive truly and I had a lot of ideas that I wanted to accomplish and achieve so I imagined and drew the design I went to carpenter and told him my idea and we started to work on it for whole of 3 days continuously to make it happen and to achieve exactly what I wanted and we faced a lot of complications along the way that had us threw the entire design and start over again and again so my part was hope I guess and hardworking on every edge of the project to come out as I've expected and better and after designing the maquette itself me and my team mates started working on the implementation of the motors on each of the doors and window which was so hard as we didn't have any background of mechanics and how does those motor works and how much torsion should be there so we just went in and tried our best really and we got it to work as best as we could and directly I started working on the flutter application with the backend with the MQTT included in it to send and receive data in real time and firebase for the authentication and then there is the UI that took me too much time because as I said before I wanted to make something different and so I watched and researched for a lot of ideas and landed on one with one of my favourite colours and I began to apply it to everything and try a lot of test cases for each and every page with the backend of each of them.

vii. The schematic of the whole project :



The flutter application I
created for this house:





[Signup](#)

[Already have an account? Sign In](#)

[Don't have an account? Sign Up](#)

Please verify your email before logging in.

10:33

◀ May lot



26



Verify your email for may-iot



Inbox



noreply 10:32 PM

to me ▾



Hello,

Follow this link to verify your email address.

[https://may-iot.firebaseio.com/_auth/action?
mode=verifyEmail&oobCode=
Wm9gcFU5SgCt8Y4rlxW7KvxDX12jJ
Y1Mrr4N5kFbugAAAGRIVLZjw&
apiKey=AIzaSyBEHWAoKRhSe-
7ZUQzzAKC6YWyx9frfX0&lang=en](https://may-iot.firebaseio.com/_auth/action?mode=verifyEmail&oobCode=Wm9gcFU5SgCt8Y4rlxW7KvxDX12jJY1Mrr4N5kFbugAAAGRIVLZjw&apiKey=AIzaSyBEHWAoKRhSe-7ZUQzzAKC6YWyx9frfX0&lang=en)

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your may-iot team

Reply

Forward



10:33

◀ May lot



26

Done

may-iot.firebaseio.com

AA



Your email has been verified

You can now sign in with your new account



5:32



Hello, How are you?

Let me start by telling you what's this application does and how it works:

1. Control all the doors and windows in your home with a click of a button.
2. Monitor all the sensor readings implemented in your house for comfort and safety.
3. Control all the lights around your home with the same click of a button.
4. View your profile and all the data you've entered during sign-up.

Hope you have a great experience with our app!

- dev.Ali Soliman



10:35



HI ALI



Smart Home

SENSORS



Fire Sensor

No Data



Gas Sensor

No Data



Water Sensor

No Data



Motion Sensor

No Data



10:35



Control Panel



Control Devices

DOORS & WINDOWS



Main Door

Open



Garage Door

Open



Window

Open

10:35



Control Panel



Control Devices

LED CONTROLS



LED 1

Turn On



LED 2

Turn On



LED 3

Turn On



5:32



Ali Soliman

alisoliman1521@gmail.com

Settings

Help & Support

About

Logout

GitHub

[Contact the developer?](#)



5:42



Sign In

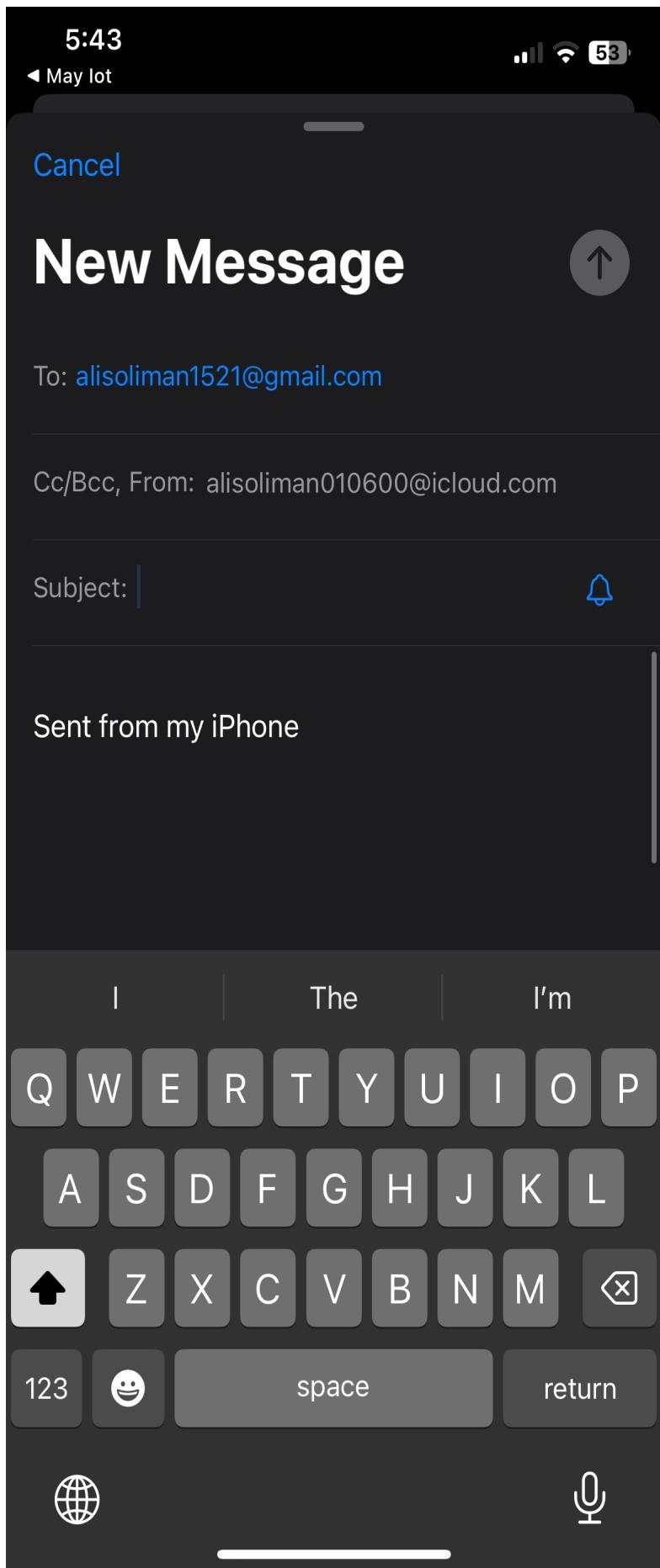
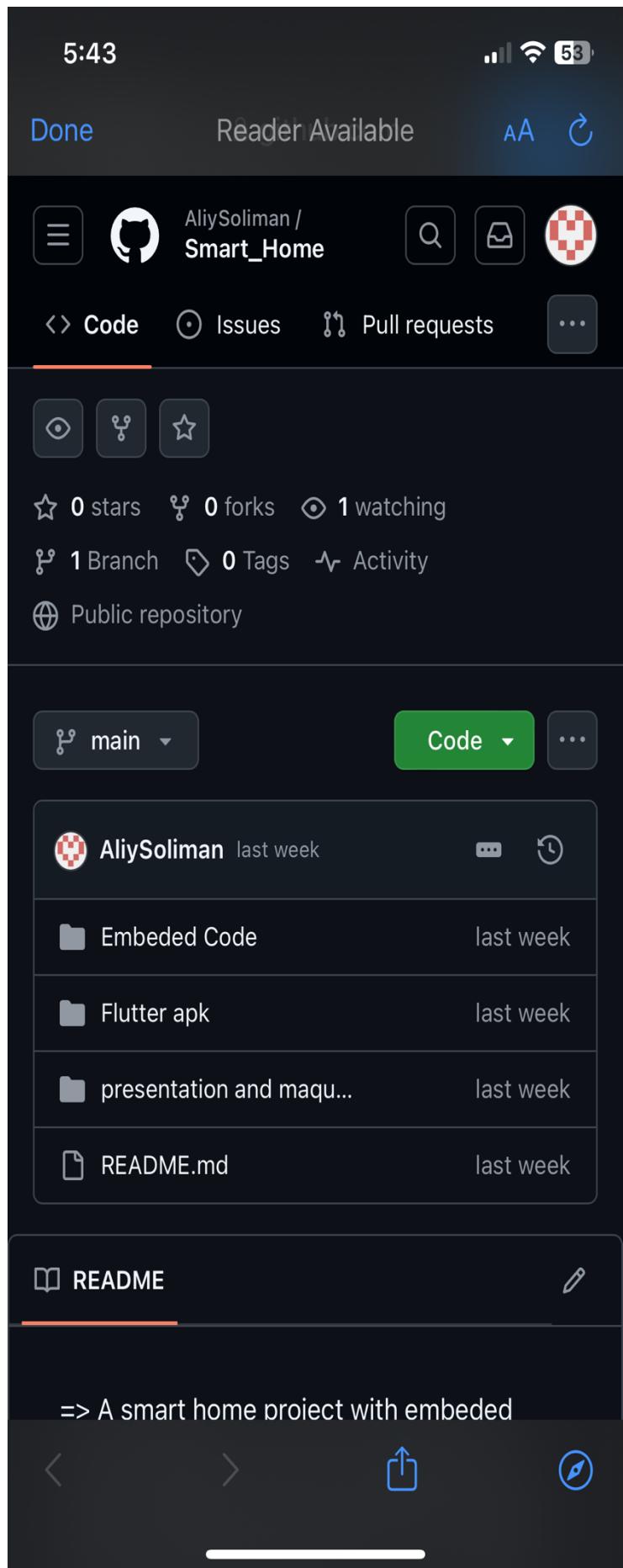
Email

Password



Login

Don't have an account? [Sign up](#) سلام سلام سلام سلام سلام



Git hub QR code



- viii. I faced a lot of challenges I even nearly gave up on this project because I didn't know where all those problems coming from but I didn't first: I had no background of designing or creating anything before so it was just an idea in my mind that I wanted to apply and I drew something but it was far from a design and I needed to work so hard on designing this maquette and being with the carpenter around the NFC all the time to see what happens and if this could work or not if this looks better would this be enough so the door can turn and I also made hinges for each of the door and window so it would be easier the servo to control and move them and then when I got to implementing the motors I had a lot to do as the first time I tried the motor didn't turn the door but the motor itself moved out of place and turned itself because the torsion between them wasn't equal and so we had to do lots of things to control the servo and help it move the doors and window with being torn apart and after all this was done there was the biggest challenge I encountered that the Esp burned out from nowhere and everything was working right the second and then it just burned and we went and bought another one and when connected we faced the same issue and I had no idea why and this Esp also burned out so we

decided to remove some of the sensors because maybe they're drawing so much power that burns the esp32 and when I went to buy another I couldn't find any left in any of the stores I looked for days didn't find any and by this time we were already late to the due time of the final project and after looking everywhere I found a store with only one left and so I went in and bought and for my surprise still didn't work and burned out by this time me and my team mated a=had lost all hope and we handed in what we could but I just couldn't after all this hard work I just wanted it to work I needed it to work and so I went online bought another and started you could say debugging connected everything on its own to see where the problem is coming from and I figured that with those modules of the Esp (divkit) it's better to connect the servos to the 3.3 volt as if connected to the Vin it draws so much power than the Esp can handle and because we are using 3 servo motors I had to threw some of the sensors away so just I could keep what's now worked after so much suffering as it is and by the discussion I'll be using 2 Esp's to connect the motors and LEDs together and the sensors together and still not all of them.

- ix. As a conclusion time is the most important thing of all as if I had encountered that challenge before I would've known how to handle it but time of my essence, I was working on all the other assignments and didn't even have time to sleep properly.

4. Chapter 4: Conclusion

- i. This training was all about hardware and microcontrollers and how everything that's built on a smart system works and how I could do it myself, so to sum up this training I did learn a lot about microcontrollers specially the Esp32 like how to use it connect other hardware to it and build an entire system with those hardware that's totally connected together and I did learn a lot about cloud and HIVEMQ with the brokers and how could the data be sent, received, monitored and controlled in real time using MQTT with the HIVEMQ which is the part the makes our system smart not just some pieces of hardware connected together and one works after another or maybe at the same time using the RTOS but rather how to connect all of them together and make them somehow dependent on each other and that's what a smart system is, and finally I learned a lot about flutter, dart and Firebase how to build an application with them and why, I also learned who to use security and AI with my Esp in the bonus sessions.
- ii. I learned a lot about microcontrollers specially the Esp32 like how to use it connect other hardware to it and build an entire system with those hardware that's totally connected together and I did learn a lot about cloud and HIVEMQ with the brokers and how could the data be sent, received, monitored and controlled in real time using MQTT with the HIVEMQ which is the part the makes our system smart not just some pieces of hardware connected together and one works after another or maybe at the same time using the RTOS but rather how to connect all of them together and make them somehow dependent on each other and that's what a smart system is, and finally I learned a lot about flutter, dart and Firebase how to

build an application with them and why, I also learned who to use security and AI with my Esp in the bonus sessions.

- iii. I explained by project in the Capstone Chapter and explained why I didn't reach my entire goal but most of it so as a future work I want to complete the entire project as I've imagined for it and I really am looking forward to apply it somehow in real life and to be a part of building a real smart home and after that I have lots of project ideas to work on using the Esp32 some of them could be added to a smart home and some are so different but still interesting to work on and who knows maybe one day you'll get to see all that I wanted to do and some of those ideas.
- iv. It already is an idea of a business and has been for a long time maybe not in Egypt but foreign countries we could say are built on smart systems especially smart homes, and so I really am looking forward to dig deep in the IOT learn more about it and apply it in real life with so extras that I want to add a develop.
- v. No other reflections.

5. References

&*ESP32 tutorials for Beginners* (no date) *ESP32 Tutorial*. Available at:
<https://esp32io.com/>.

- i. *IDF Programming guide* (no date) *ESP*. Available at:
<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html#>.
- ii. Cameron, N. (2021) *Electronics projects with the ESP8266 and ESP32: Building web pages, applications, and WIFI enabled devices*. Berkeley, CA: Apress L.P.
- iii. Tuan-Karma (no date) *Tuan-Karma/ESP32_WiFiClientSecure: Copy of the 'https://github.com/espressif/arduino-esp32' s wifiClientSecure library for fixing bugs and improments, GitHub*. Available at: https://github.com/tuan-karma/ESP32_WiFiClientSecure (Accessed: 11 September 2024).

VI. Feedback & Reflection

- a. Most of the course were full of positive aspects except for the timing cycle and the communications.
- b. For next Cycle:
 - I. Same training content but include the AI and Security sessions in the real schedule and added to the content not to be considered bonus sessions.
 - II. 2 moth maybe more if AI and security were added to the content.

III. Training Methods:

1. All lectures should be offline with a follow up session 2 or 3 days after for questions and more practice.
2. Workshop was good also the number of students was so much to be handled and supervised equally so I would like if the numbers should go down a bit maybe half of this year's numbers per session, it would be much better supervised and even easier for both the students and constructors to handle.
3. On-site practice was pretty good.
4. Assignments were pretty good too.
5. The project should be introduced and talked about much sooner so we'd have enough time to do it and handle every error and issue that could be encountered.

IV. Were pretty good too.

V. They're very good and helpful.

VI. It wasn't that good but that was because there were so many of us asking a lot of questions at the same time so it's more of a numbering per course problem.

VII. Was good.

c. No other businesses.

V. Acknowledgment

a. Lots and lots of research and YouTube videos.