

**FLAREPATH – ADVANCED VEHICLE FIRE SAFETY
AND MONITORING WITH RAPID EMERGENCY
DISPATCH SOLUTIONS**

R24-058

Status Document-1



Dharmagunawardana W.M.P.I – IT21132346

B.Sc. (Hons) Degree in Information Technology specializing in

Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

May 2024

Group Details

Supervisor – Mr.Nelum Chathuranga Amarasena

Co-supervisor – Mr. Deemantha Nayanajith Siriwardana

External Supervisor – Mr. Onray Sahinda

Student Name	Student ID	Contact No	Email Address
Dharmagunawardana W.M.P.I	IT21132346	0772785361	it21132346@my.sliit.lk
Anthick G.N	IT21096266	0779820516	it21096266@my.sliit.lk
Abeywardhana D.N	IT21133718	0714057155	it21133718@my.sliit.lk
Peramunage A.N	IT21080562	0713999266	it21080562@my.sliit.lk

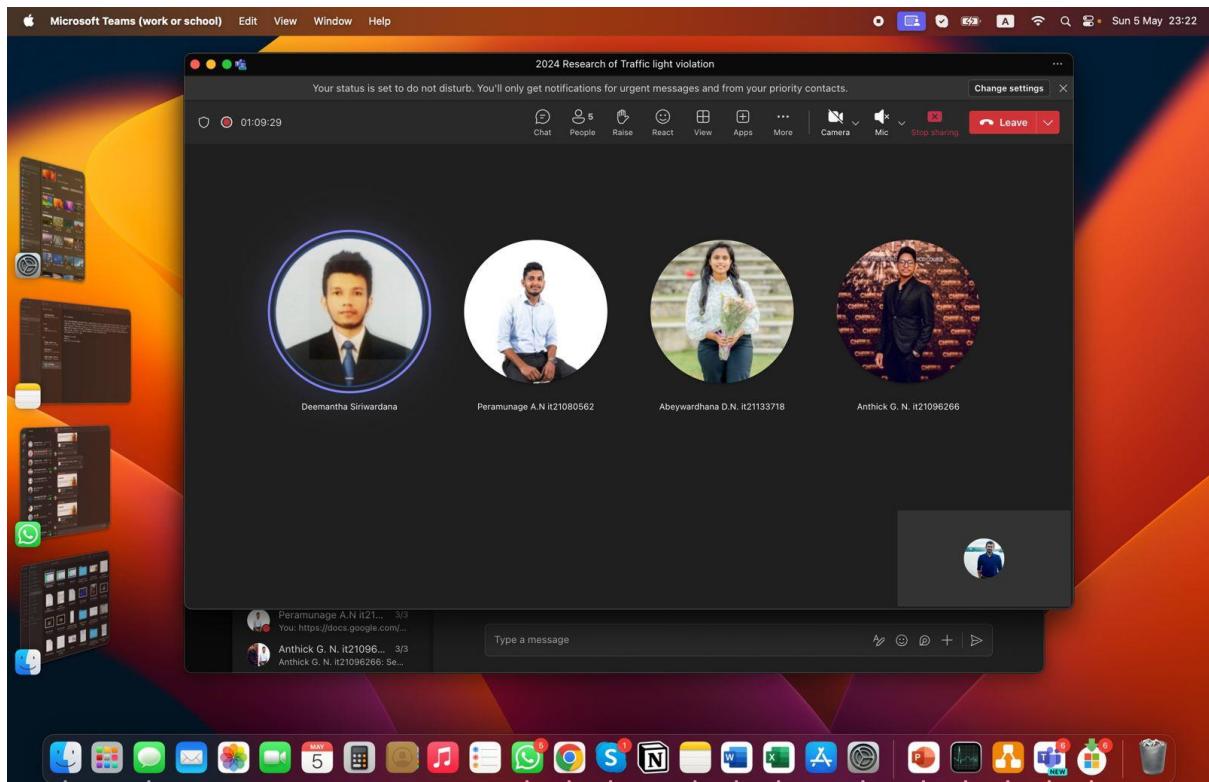
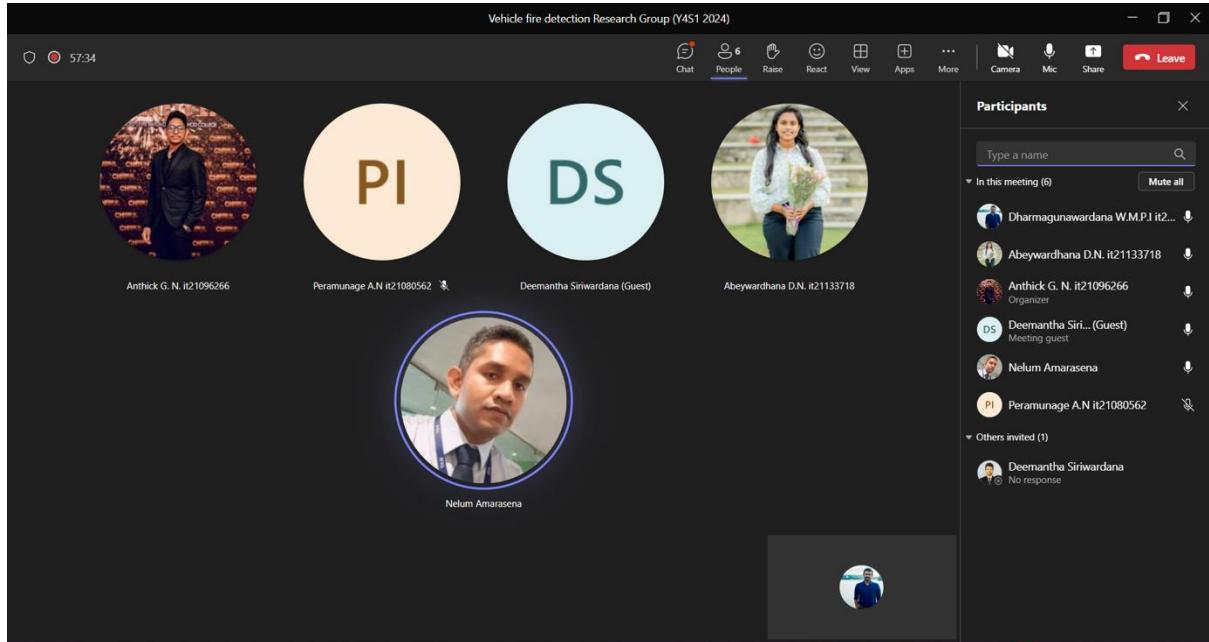
Table of Contents

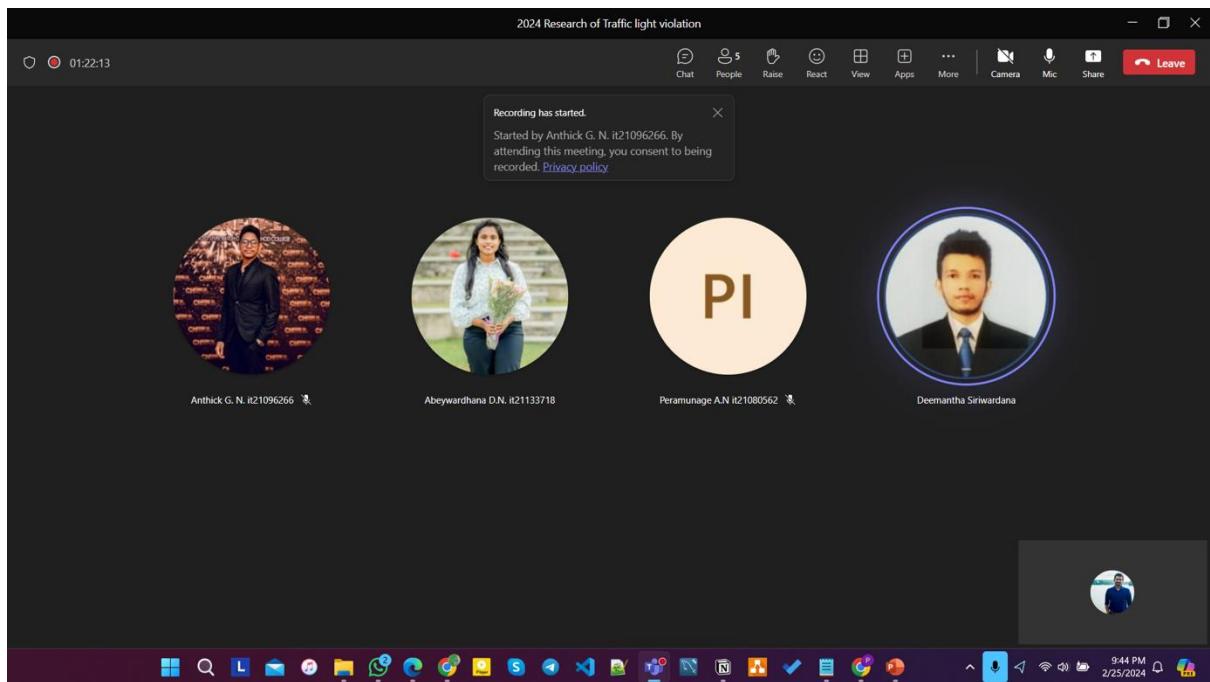
Group Details	2
Meetings & Calls.....	4
Meetings with supervisor and co-supervisor	4
Meetings with Domain Experts	8
Snapshots from Field Visit.....	9
Click Up Tasks Allocation	11
Click Up Dashboard	11
In Progress Tasks	12
Completed Tasks up to PP1.....	12
Project Implementation	13
Data Collection	13
IOT Based Fire Extinguisher Mechanism	14
IOT Device Code Base	14
Results after compilation.....	19
IOT Device Implementation.....	20
Wireframes	21
Mobile Application UI	23
Gantt Chart	24
Work Breakdown Structure	24

Meetings & Calls

Meetings with supervisor and co-supervisor

Meeting with both supervisor and co-supervisor about the project progress and improvements that we need to do to our project.





2024 Research of Traffic light v... Chat Files Recap Speaker Coach Q&A + Join 0:0 5 : ...

Anthick G. N. it21096266 1/5 11:39 PM

Hi sir, can we arrange a meeting within next two days if you can. We have found some research papers and got some of the components. so we can discuss if there any issue in that, since we have to submit TAF on 11th this month

January 7

You're invited to Co-Supervisor meeting
Sunday, 7 January 2024
9:30 pm - 10:30 pm (GMT+5:30)
https://teams.microsoft.com/l/meetup-join/19%3ameeting_mDc3YmEtMmltNDQxYS00MzEyLTkYjgtMiVmNGl3MzY4NjQy%40thread.v2/0?context=%7b%22Id%22%3a%2244e3cf94-19c9-4e32-96c3-14f5bf01391a%22%2c%22Oid%22%3a%22b3aefc1cb656-4424-ad90-3b6463f1a17e%22%7d
Tap on the link or paste it in a browser to join.

Join conversation teams.microsoft.com

1/7 9:17 PM Meeting ended: 14s

1/7 9:17 PM Meeting started

1/7 10:02 PM Recording has started

1/7 10:13 PM Meeting ended: 56m 15s

1/7 10:13 PM Recording has stopped. Saving recording...

Co-Supervisor meeting Recording Abeywardhana D.N. it2... 11m 53s

This recording is set to expire. View or change the expiration date [here](#). [Learn more](#)

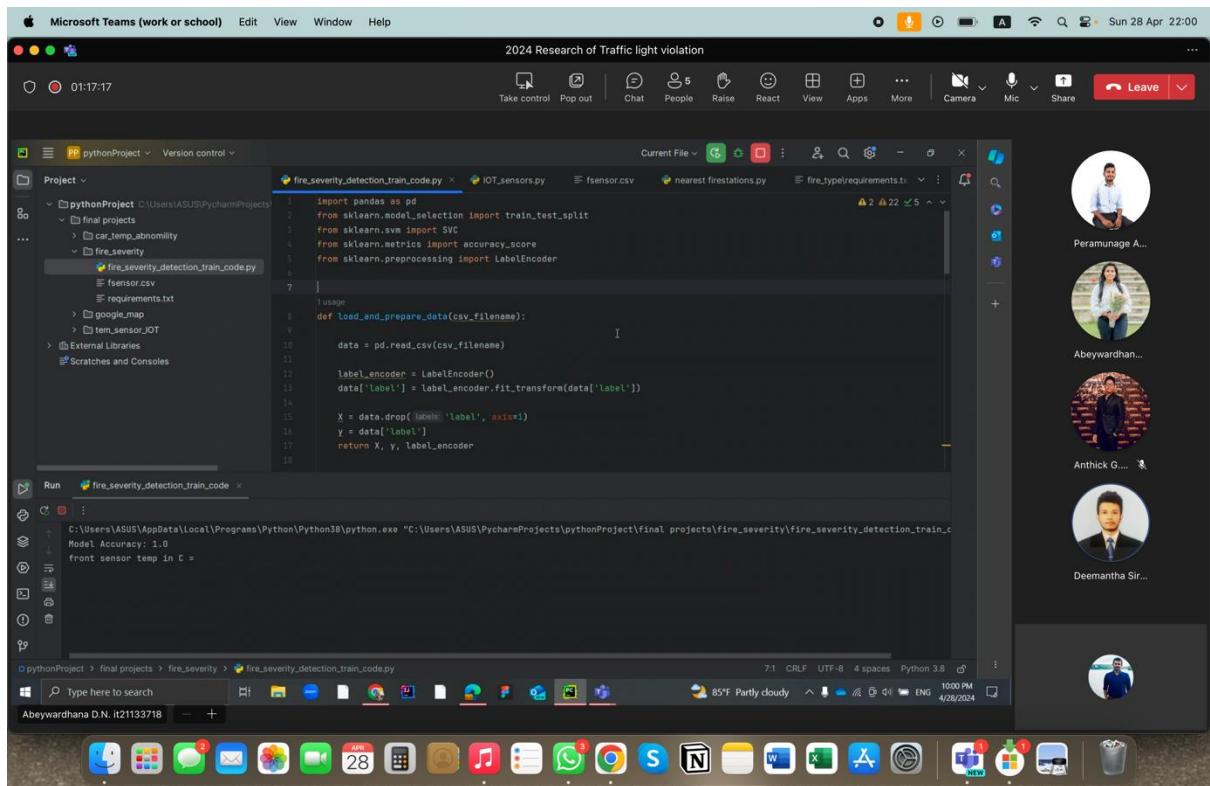
January 16

1/16 9:48 PM Meeting ended: 14s

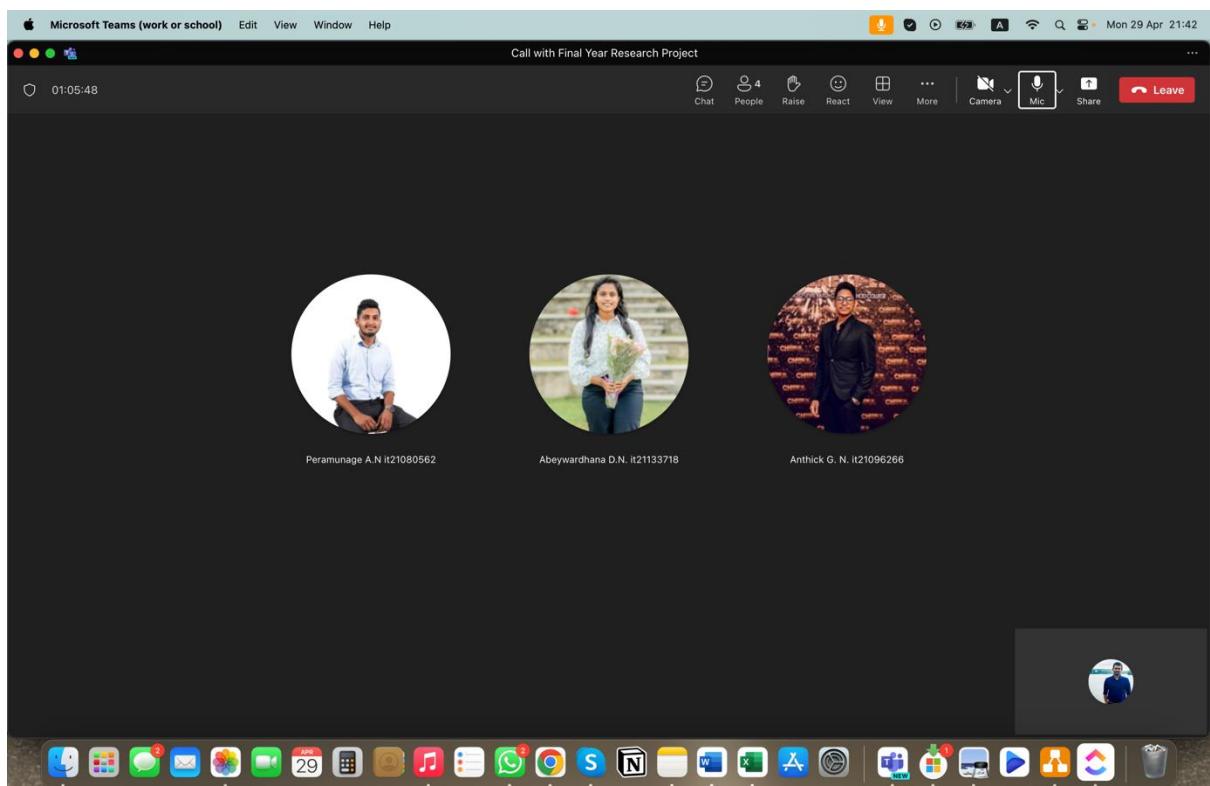
January 17

Type a message

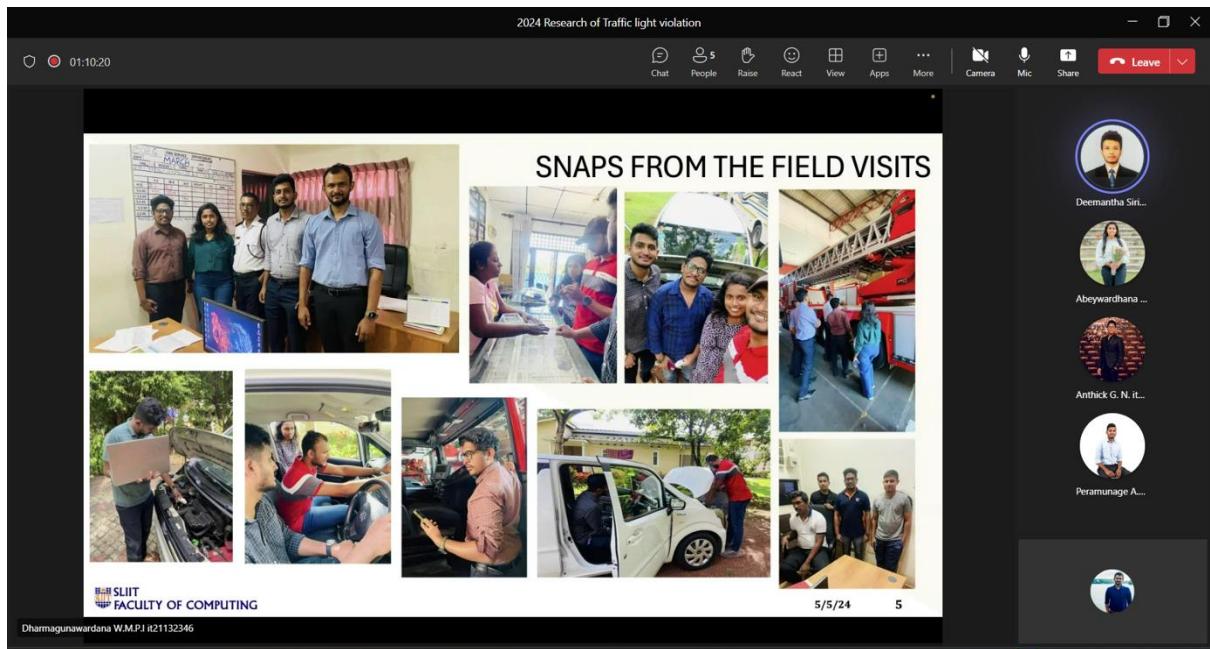
Code review with co-supervisor



Group meeting with group members



Presentation review with supervisor



Meetings with Domain Experts

Meeting with fire department officers and staff.

Mr.Nanayakkara the chief officer of the fire department and we discussed about the domain knowledge and requirements.



	january	february	march	april	may	june	july	august	september	october	november	december	TOTAL
FIRE CALL	38	31	48	10	24	12	25	25	24	21	16	25	299
RESCUE CALL	2	1	5	2	1	2	0	2	4	2	3	2	26
EMERGENCY CALL	6	1	0	1	8	0	0	3	2	9	2	5	37
AMBULANCE CALL	0	1	2	0	2	0	1	2	0	1	6	9	24
VIP DUTIES	37	37	45	1	1	2	15	33	32	38	32	33	306
SPECIAL SERVICE	24	19	12	11	2	7	1	6	6	5	10	20	123
TEST CALL	3	3	3	0	1	5	1	2	5	5	7	9	44
INSPECTION OF DANGER PLASE	0	0	0	0	0	0	0	0	0	39	16	40	95
TOTAL	110	93	115	25	39	28	43	73	73	120	92	143	954

prepared by - K.T.S.Fernando

K.P.P.R Nanayakkara
Control Room officer

A.P.J.Preethilal
Station officer
(Communication)

W.S.R.N Senanayake
Divisional fire officer
(Operation)

P.D.K.A.Wilson
Chief fire officer

Snapshots from Field Visit





Click Up Tasks Allocation

The screenshot shows a Click Up board titled "Research - 058". The board has several columns representing different stages of task progression:

- To Do:** 0 tasks
- Planning:** 4 tasks
 - Implement the Fire sensor
 - Create a mobile app
 - Add few more sensors and increase the accuracy
 - Fire severity model creation
- In Progress:** 11 tasks
 - create a mobile app
 - Decide features
 - Create Figma design
 - Fire severity data set creation
 - mobile app Design
 - Train the model
- Complete:** 13 tasks
 - Uploaded a Blinking program to the ESP32
 - Arduino IDE download and setup
 - Fire sensor Integration with ESP32
 - optimal path suggestion
 - nearest fire department identification
 - Fire department location data collection and data set creation
- Cancelled:** 0 tasks

Each task card includes details like status, assignee, and due date.

Click Up Dashboard

The screenshot shows a Click Up dashboard titled "Dashboard". It displays the following key metrics:

- Unassigned:** 10 tasks
- In Progress:** 15 tasks in progress
- Completed:** 13 tasks completed
- Tasks Completed This Week:** No results

Below these metrics are two visualizations:

- Total Tasks by Assignee:** A pie chart showing the distribution of tasks among assignees. The largest segment is "Unassigned" at 44.11%, followed by "it21132346 Dharma..." at 20.00%, "it21096266 Anithick G. N." at 17.64%, "it21080562 Peramunage A.N." at 8.82%, and "it21133718 Abeywardhana D.N." at 8.82%.
- Open Tasks by Assignee:** A bar chart showing the number of open tasks for each assignee. The bars are labeled with assignee names: "it21132346 Dharma...", "it21096266 Anithick G. N.", "it21080562 Peramunage A.N.", "it21133718 Abeywardhana D.N.", and "Unassigned".

On the right side of the dashboard, there is a section for "Latest Activity" which shows a single entry: "Fire severity model creation" by "it21132346 Dharma..." on "Today".

In Progress Tasks

The screenshot shows a ClickUp dashboard titled "In Progress". It displays 15 tasks in progress, each with an icon, title, assignee, due date, and status. The tasks are:

- Implement the Fire sensor
- create a circuit Diagram
- create a mobile app [t_2]
- Fire severity data set creation
- mobile app Design [t_2]
- Train the model
- Fire prediction data set creation [t_0]
- fire Prediction model
- fire prediction data collection
- fire department resource allocation prediction with the severity of the fire.
- Web Application Wireframes Design
- Add few more sensors and increase the accuracy
- fire prediction monitoring IOT device [t_0]
- Mobile Application Design
- Fire severity model creation

The status column indicates 10 tasks are "IN PROGRESS" and 5 are "PLANNING".

Completed Tasks up to PP1

The screenshot shows a ClickUp dashboard titled "Completed". It displays 13 tasks completed, each with an icon, title, assignee, due date, and status. The tasks are:

- Uploaded a Blinking program to the ESP32
- Arduino IDE download and setup [t_0]
- Fire sensor Integration with ESP32
- optimal path suggestion
- nearest fire department identification
- Fire department location data collection and data set creation
- fire department resource data collection
- Draw Swimlane chart
- Create a swimlane diagram
- Web application Wireframes Design [t_0]
- Swimlane chart design
- Web Application Wireframes Design
- Fire department data collection

The status column shows all 13 tasks as "COMPLETE".

Project Implementation

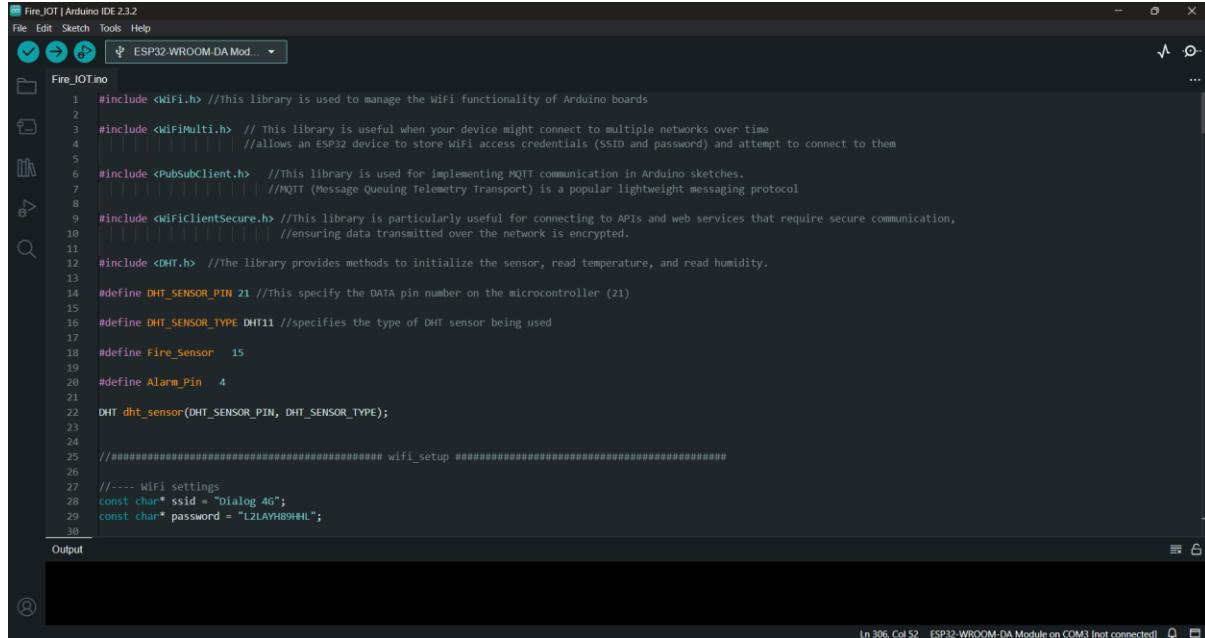
Data Collection

Collecting the temperature and RPM data using the thermometer and a RPM gauge.



IOT Based Fire Extinguisher Mechanism

IOT Device Code Base



```
#include <WiFi.h> // This library is used to manage the WiFi functionality of Arduino boards
#include <WiFiMulti.h> // This library is useful when your device might connect to multiple networks over time
#include <PubSubClient.h> // This library is used for implementing MQTT communication in Arduino sketches.
#include <WiFiClientSecure.h> // This library is particularly useful for connecting to APIs and web services that require secure communication,
//ensuring data transmitted over the network is encrypted.

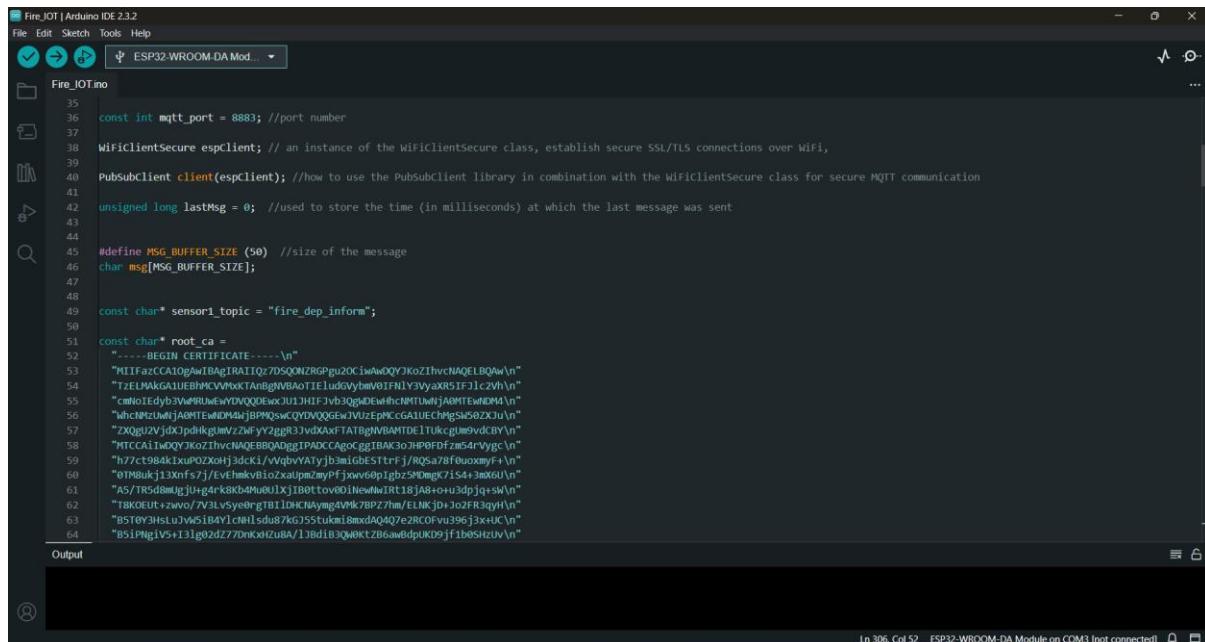
#define DHT_SENSOR_PIN 21 //this specify the DATA pin number on the microcontroller (21)
#define DHT_SENSOR_TYPE DHT11 //specifies the type of DHT sensor being used
#define fire_Sensor 15
#define Alarm_Pin 4

DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

//#####
// WiFi settings
const char* ssid = "Dialog 4G";
const char* password = "L2LAyH89HHL";
//#####

Output
```

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]



```
const int mqtt_port = 8883; //port number

WiFiClientSecure espclient; // an instance of the WiFiClientSecure class, establish secure SSL/TLS connections over WiFi,
PubSubClient client(espclient); // how to use the PubSubClient library in combination with the WiFiClientSecure class for secure MQTT communication
unsigned long lastMsg = 0; //used to store the time (in milliseconds) at which the last message was sent

#define MSG_BUFFER_SIZE (50) //size of the message
char msg[MSG_BUFFER_SIZE];

const char* sensor1_topic = "fire_dep_inform";

const char* root_ca =
"-----BEGIN CERTIFICATE-----\n"
"MIIFazCCA0IjAwBAGtRAIQQz7DSQNZRGPguZ0CiAwDQYJKoZIhvNaQEJBQAw\n"
"TELMakGAUEBhKCVWgkTAngNVAAoTIEluoVbybm01NlY3vyaxR5IfJlc2Vh\n"
"cmtoIEdydVwURUwevYDVQDXEwJU1HfTJvb3QgADEwInNNTUwA0MTExwDM4\n"
"WhcNzUwlijAOETEW0DMA4iBPMoswQYDVQGeEwJUzphCCALUEChlgS4s02Xh\n"
"ZxQgUzVjdxjpdhkgmVzwyY2ggj33vdXAxFTAT1BgNBANTDE1UkcgMsvdGBV\n"
"MTCA1IwQYJKoZIhvNaQEBOADgTPADCCAgOcgTBAdjHPDFdz54rvyc\n"
"h77ct984kxu02x0j3dcK1jVqbxvAtYjb3m1geSttrfj/QS5a7f8fouoxmyF\n"
"01Nbukj13Xmf57jEvHmkvBio/xalpmzmyFjxw60p1gb5f0mgx7154+3mXG\n"
"AS/TSRsdm0jUj0+4rK8kb4Mu0U(x)jB0tov0d1newdwIR18[A8]+h+3dpjq+s\n"
"TR8QELutzwvo/7V3LvsyeengTB11HCNAVmp4Mk7BZPZ7m/E1NKjd+1oZR3qy\n"
"B5T0Y3HSLujW518AY1cNrlsdub7kgJ55tukm18mxQ4Q7e2RCOFvu396j3x+UC\n"
"B5iPNgi1v5+131g02d277DnKhZu8A/1Bd183QWktZB6aw8dpUKD9jf1b0SHzU\n"
"Vn"
```

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]

Fire_IOT | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32-WROOM-DA Mod... ▾

Fire_IOT.ino

```
63
64 bool msgSent = false;
65
66 // Initializes serial communication, sets up WiFi connection using SSID and password, and connects to WiFi.
67 void MQTT_client_setup() {
68
69     delay(5000); // pauses the execution of the program for 5000 milliseconds, or 5 seconds
70
71     Serial.begin(115200); // set the informations transfer rate to 115200 / 115200 is one of the standard baud rates
72     Serial.print("\nConnecting to ");
73     Serial.println(ssid); // prints the ssid name of the connection
74
75     WiFi.mode(WIFI_STA);
76     WiFi.begin(ssid, password);
77
78     // This loop continuously checks the WiFi connection status
79     while (WiFi.status() != WL_CONNECTED) {
80         delay(500); // it waits for 500 milliseconds and prints a dot (.) on the serial monitor
81         Serial.print(".");
82     }
83
84     randomSeed(micros()); // Initializes the pseudo-random number generator
85     Serial.println("\nWiFi connected\nIP address: ");
86     Serial.println(WiFi.localIP()); // Prints the IP address assigned to the device on the network.
87
88     while (!Serial) delay(); // is a loop that continues as long as Serial is false
89
90     espClient.setCACert(root_ca); // CA certificate is used to verify the certificate of the server the device is connecting to
91     espClient.setSSLCert(root_ca); // for client verification
92 }
```

Output

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected] ▾

Fire_IOT | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32-WROOM-DA Mod... ▾

Fire_IOT.ino

```
114 client.setServer(mqtt_server, mqtt_port); // configures the MQTT client with the server (broker) address and the port it will connect to for MQTT
115 client.setCallback(callback); // This method sets the callback function that the MQTT client will call whenever it receives a message
116
117 // to manage the WiFi connection status within an IoT device
118 if ((WiFi.status() != WL_CONNECTED)) { // Checks the current status of the WiFi connection.
119
120     Serial.print(millis()); // Prints the number of milliseconds
121
122     Serial.println("Reconnecting to WiFi..");
123     WiFi.disconnect(); // Disconnect from the wifi
124     WiFi.reconnect(); // attempt to reconnect to the wifi
125
126 }
127
128
129
130 //Interrupt service interrupt to other functions and call alarm function
131 void IRAN_ATTR Fire()
132 {
133     alarm();
134
135 }
136
137
138 ////////////////////////////////////////////////////////////////// after connect function MQTT /////////////////////////////////
139
140
141 void reconnect() { //reconnect Function starts here
142
143 }
```

Output

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected] ▾

```
Fire_IOT | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32-WROOM-DA Mod...
Fire_IOT.ino
139
140
141 void reconnect() { //reconnect Function starts here
142
143     // Loop until we're reconnected
144     while (!client.connected()) { // Initiates a loop that continues as long as the MQTT client is not connected.
145         Serial.print("Attempting MQTT connection...");
146         String clientId = "esp-client-"; // Creates a unique client ID for the MQTT connection.
147         boolean cleanSession = true;
148         clientId += String(random(0xffff), HEX); // generates a random number between 0 and ffff
149
150
151         // Attempts to connect to the MQTT broker using the generated client ID and stored credentials.
152         if (client.connect(clientId.c_str(), mqtt_username, mqtt_password)) {
153
154             Serial.println("connected");
155
156             client.subscribe(sensor1_topic); // subscribe the topics here
157
158         } else { // Provides feedback and manages actions if the connection attempt fails.
159             Serial.print("failed, rc=");
160             Serial.print(client.state()); // prints display the connection failure reason with an error code
161             Serial.println(" try again in 5 seconds"); // Wait 5 seconds before retrying
162             delay(5000); // pauses the loop for 5000 milliseconds (5 seconds)
163         }
164     }
165 }
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
```

Output

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]

```
Fire_IOT | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32-WROOM-DA Mod...
Fire_IOT.ino
167
168
169 void callback(char* topic, byte* payload, unsigned int length) { // Defines the function that the MQTT client will call upon receiving a message.
170
171     String incomingMessage = ""; // initializes an empty string to store the incoming message
172
173     for (int i = 0; i < length; i++) incomingMessage += (char)payload[i]; // Converts the payload from an array to a string.
174
175     Serial.println("Message arrived [" + String(topic) + "] " + incomingMessage); // Logs the received message along with its topic
176
177     String intopic = String(topic); // Converts the topic from a C-style string (char*) to an Arduino String object.
178
179     if_message_resv(intopic, incomingMessage); // Further processes the received message by calling another function
180
181
182
183 //===== publishing as string
184 void publishMessage(const char* topic, String payload, boolean retained) { // Defines a function to publish messages to an MQTT topic
185
186     if (client.publish(topic, payload.c_str(), true)) // Attempts to publish a message to the specified MQTT topic.
187
188         Serial.println("Message published [" + String(topic) + "]: " + payload); // Logs the successful publication of the message.
189
190
191
192
193 void wifi_down_check() { // function to check and manage the WiFi and MQTT connection status.
194
195     Serial.println("entering the loop");
196
```

Output

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]

```
void wifi_down_check() { // function to check and manage the WiFi and MQTT connection status.
    Serial.println("entering the loop");
    // if WiFi is down, try reconnecting every CHECK_WIFI_TIME seconds
    if ((WiFi.status() != WL_CONNECTED)) { //checks if the device is not connected to WiFi.
        Serial.print(millis()); // providing a timestamp for when the reconnection attempt happens.
        Serial.println("Reconnecting to WiFi...");
        WiFi.disconnect(); // disconnects from the WiFi network. This is useful if the connection is in a bad state and needs resetting.
        WiFi.reconnect(); // Attempts to reconnect to the WiFi network
    }
    if (!client.connected()) reconnect();
    client.loop(); // Maintains the MQTT client connection,
}
//######################################## after connect function MQTT END #####
float temp_sensor(int annotation = 1){ //Defines a function named temp_sensor that optionally annotates readings by printing them.
// int annotation: A parameter with a default value of 1, which controls whether the sensor readings are printed to the serial monitor.
// If set to 1, readings will be printed.
// Basic humidity and temperature from a DHT sensor
}
```

```
//Reads humidity and temperature from a DHT sensor.
float humi = dht_sensor.readhumidity();
float tempC = dht_sensor.readtemperature();
float tempF = dht_sensor.readtemperature(true);
if (annotation == 1){
    // Validates the sensor readings.
    if (isnan(tempC) || isnan(tempF) || isnan(humi)) { // checks each reading if it's "Not a Number" (NaN), which indicates a failed reading from the sensor.
        Serial.println("Failed to read from DHT sensor!");
    } else {
        // Outputs the humidity and temperature readings to the serial monitor.
        Serial.print("Humidity: ");
        Serial.print(humi);
        Serial.print("%");
        Serial.print(" | ");
        Serial.print("Temperature: ");
        Serial.print(tempC);
        Serial.print("°C ~ ");
        Serial.print(tempF);
        Serial.print("°F");
    }
}
return tempC;
}
```

Fire_IOT | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32-WROOM-DA Mod...

Fire_IOTino

```
250
251 //##### if msg reseved then do #####
252 // This void is called every time we have a message from the broker
253
254 void if_message_resv(String topic, String msg) {
255
256     //outputs the received message along with the topic to the serial monitor.
257     Serial.print("serial workssssssssssss - ");
258     Serial.print(topic); //print the topic of the serial
259     Serial.print(" - ");
260     Serial.println(msg);
261 }
262 //#####
263
264 // Main Functions, all the arduino logics will call through this functions / setup() & loop() functions
265 void setup() {
266
267     MQTT_client_setup();
268     dht_sensor.begin(); //initializes the DHT sensor for reading temperature and humidity data.
269
270     pinMode(Fire_Sensor, INPUT_PULLUP);
271
272     attachInterrupt(Fire_Sensor, Fire, RISING); //Rising edge -button push
273
274     pinMode(Alarm_Pin, OUTPUT);
275
276 }
277
278 }
```

Output

Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]

Fire_IOT | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32-WROOM-DA Mod...

Fire_IOTino

```
280
281 void loop() {
282
283     wifi_down_check(); // function call that continuously checks the wifi and attempt to reconnect
284
285     /////////////////
286
287     float room_temp = temp_sensor(e);
288
289     Serial.print("sensor temp = ");
290     Serial.println(room_temp);
291
292     dtostrf(room_temp, 4, 2, msg); // Converts the floating-point number room_temp into a string format to have a width of 4 characters and 2 decimal places.
293     // The result is stored in msg, which is an array of characters.
294
295     Serial.println("Publishing Message ");
296     publishMessage(sensor1_topic, msg, true);
297     delay(1000); //Pauses the execution of the loop() function for 1000 milliseconds (or 1 second), creating a delay between each execution of the loop.
298
299 }
300
301 void alarm(){
302
303     Serial.println("ALARM BUZZER");
304     Serial.println(digitalRead(Alarm_Pin));
305     digitalWrite(Alarm_Pin, !digitalRead(Alarm_Pin));
306
307
308 }
309 }
```

Output

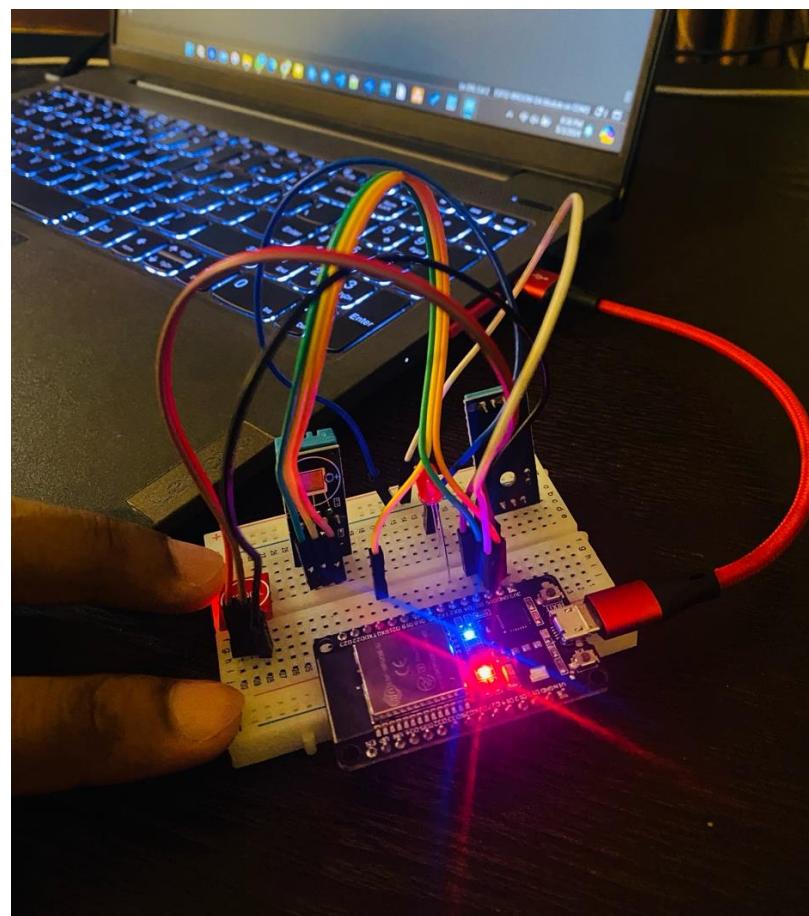
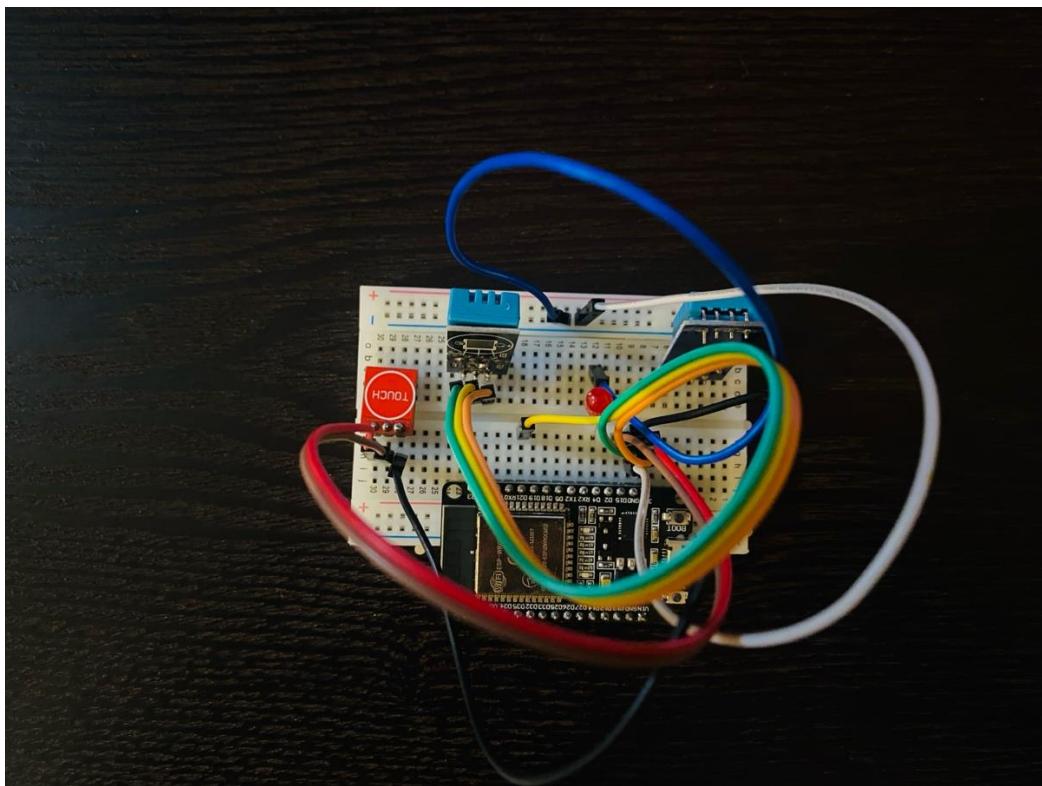
Ln 306, Col 52 ESP32-WROOM-DA Module on COM3 [not connected]

Results after compilation

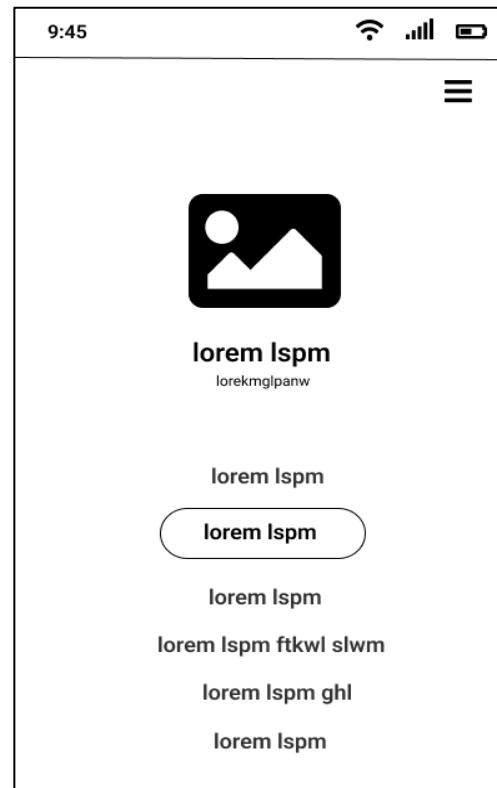
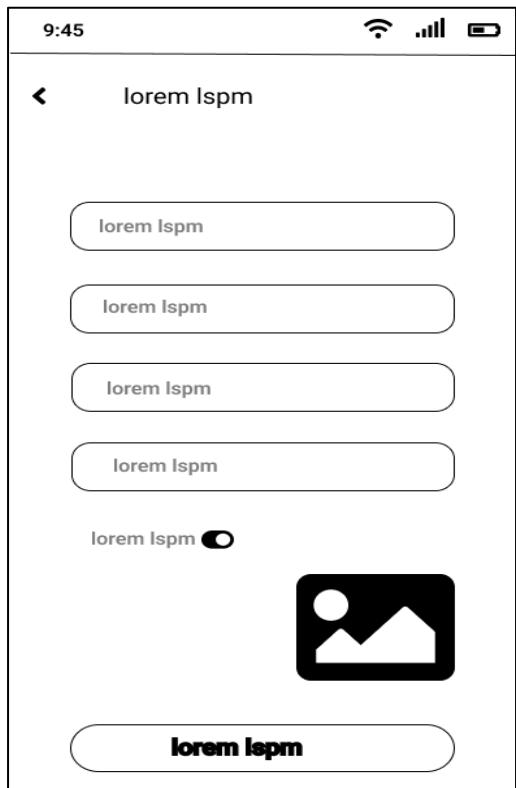
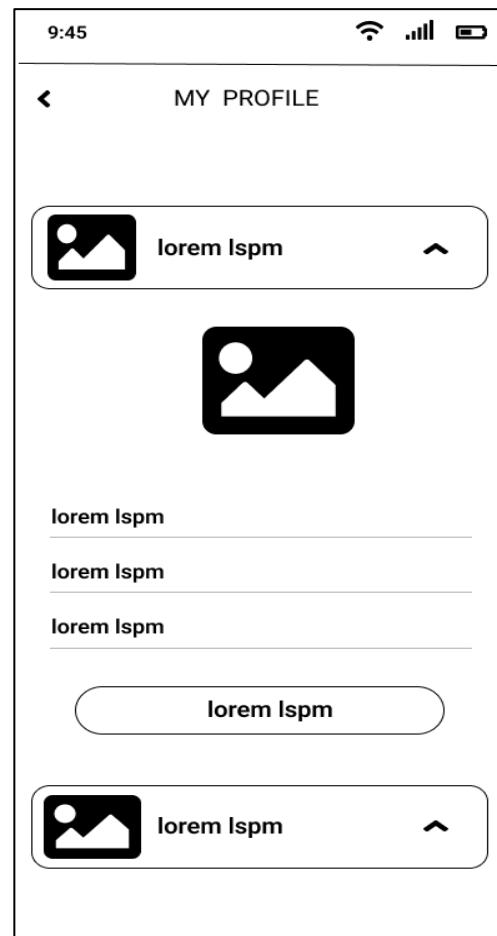
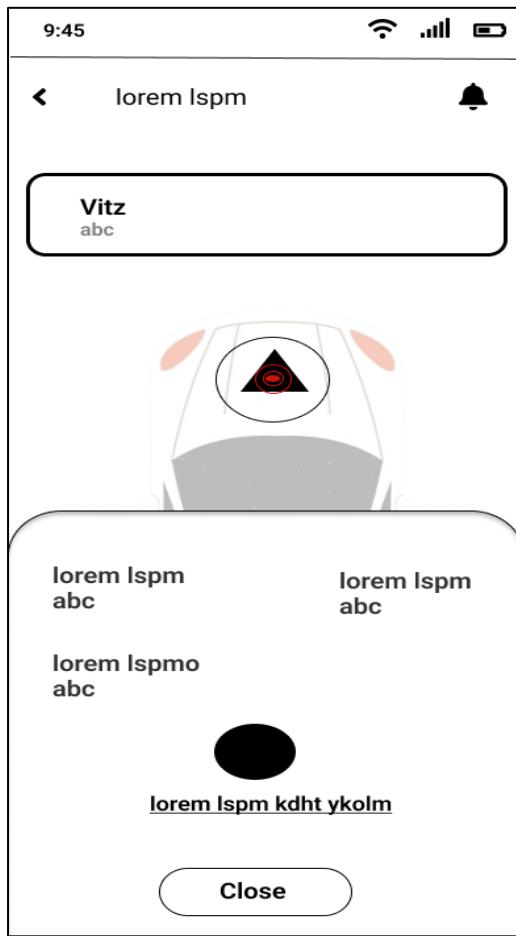
```
File Edit Sketch Tools Help
Fire_IoTino
ESP32-WROOM-DA M...
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')
New Line 115200 baud

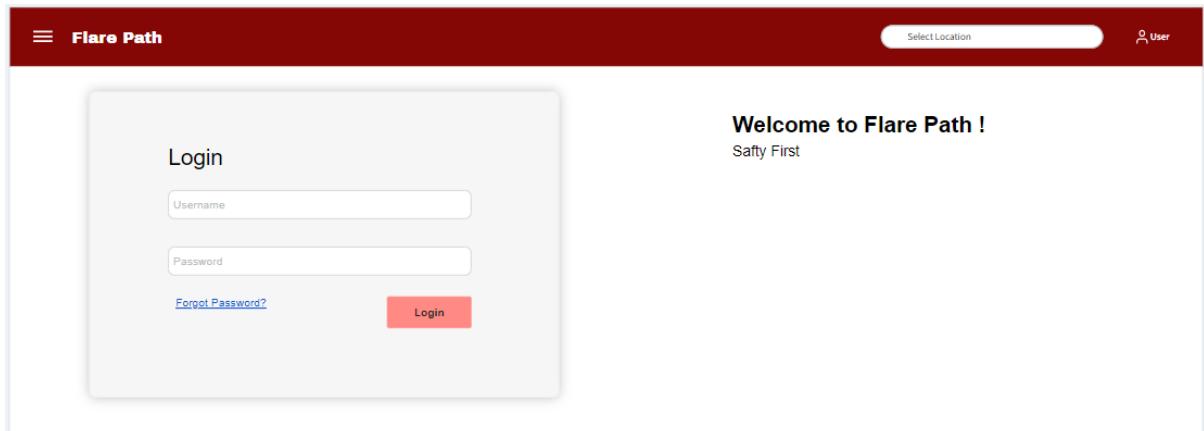
ALARM_BUZZER
0
ALARM_BUZZER
1
ALARM_BUZZER
0
ALARM_BUZZER
1
ALARM_BUZZER
0
ALARM_BUZZER
1
entering the loop
Message arrived [fire_dep_inform] 31.30
serial workssssssssssss - fire_dep_inform - 31.30
sensor temp = 31.30
Publishing Message
Message published [fire_dep_inform]: 31.30
ALARM_BUZZER
0
ALARM_BUZZER
1
ALARM_BUZZER
0
ALARM_BUZZER
1
entering the loop
Message arrived [fire_dep_inform] 31.30
serial workssssssssssss - fire_dep_inform - 31.30
sensor temp = 31.30
Publishing Message
Message published [fire_dep_inform]: 31.30
```

IOT Device Implementation

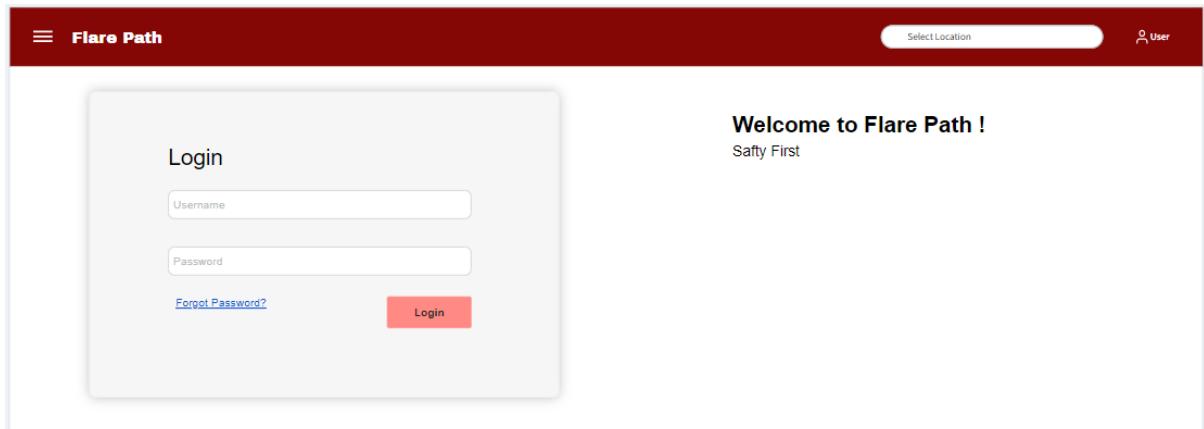


Wireframes

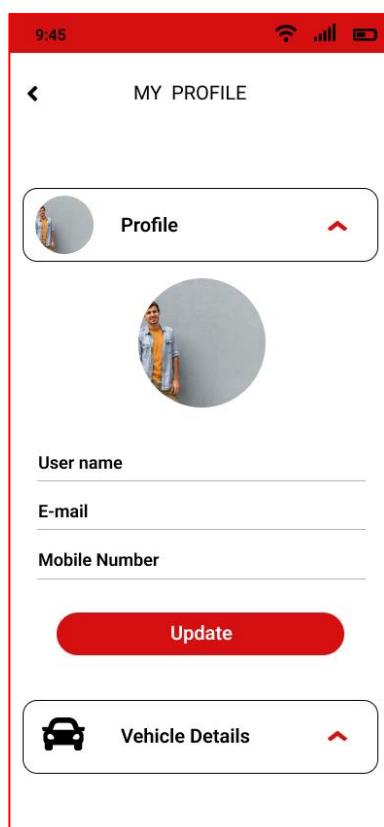
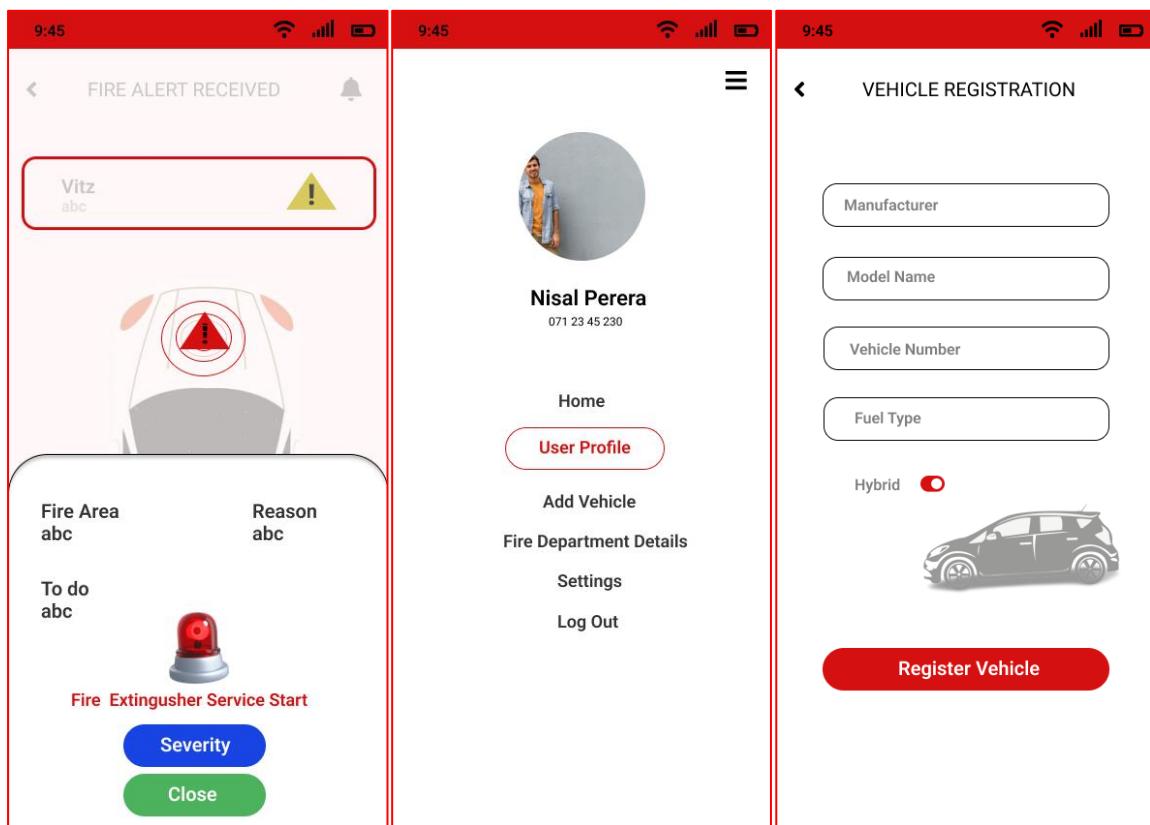




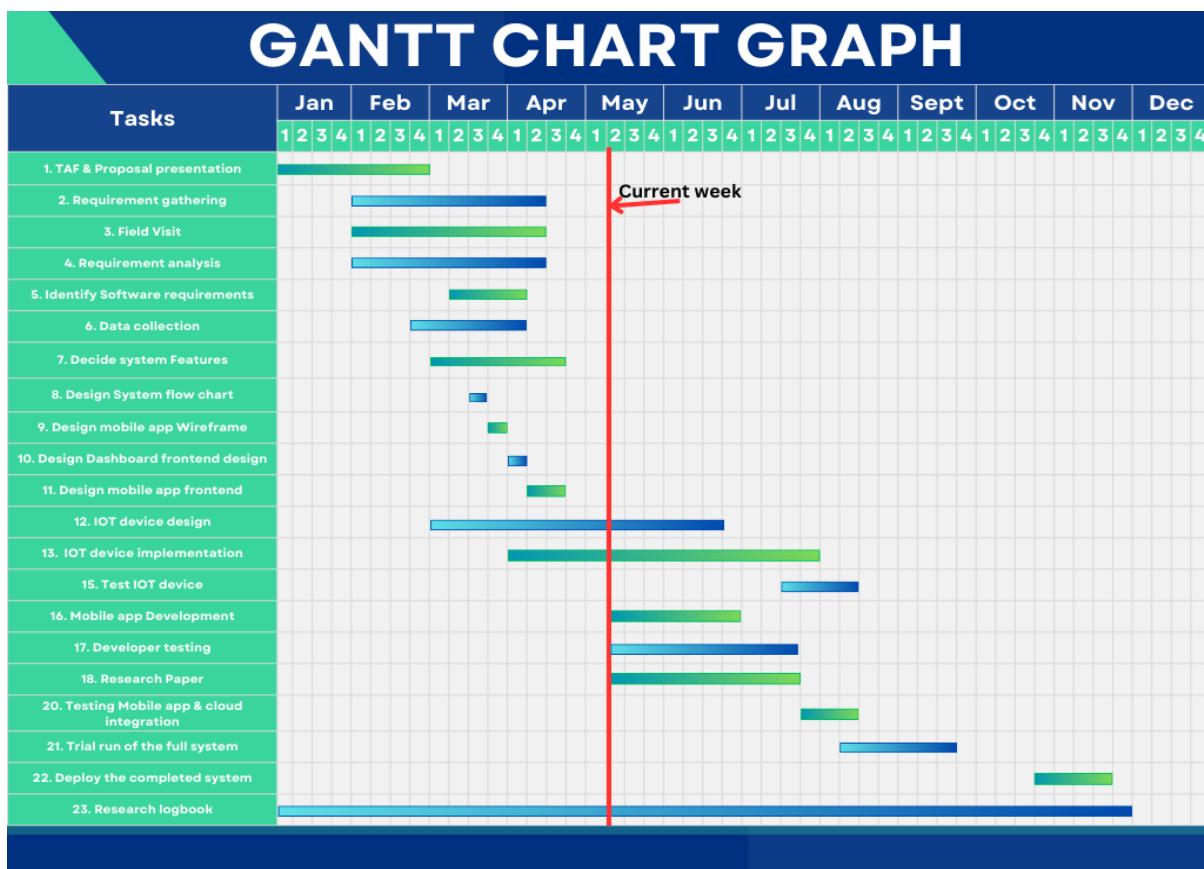
This screenshot displays the "User Role" management page within the Flare Path application. The left sidebar has a "User Management" section with "User Roles" selected. The main content area is titled "User Role". It contains several input fields: "User Role" (placeholder: Placeholder), "Select Location" (two dropdown menus with "X" buttons), and "Description" (placeholder: Placeholder). Below these are "Password" and "Confirm Password" fields, both with placeholder text "Placeholder". A "Create User Role" button is located to the right. At the bottom, there's a table with columns "Role", "Location", "Description", and "Action". The table lists "Super Admin" and "Admin" roles, each with four edit icons in the "Action" column.



Mobile Application UI



Gantt Chart



Work Breakdown Structure

