

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

R24-058

Status Document - 2



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

September 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.....	9
Snapshots from Field Visit.....	Error! Bookmark not defined.
Click Up Tasks Allocation	10
Click Up Dashboard	10
In Progress Tasks	Error! Bookmark not defined.
Completed Tasks up to PP1	11
Project Implementation	12
Data Collection	12
IOT Based Fire Extinguisher Mechanism.....	14
IOT Device Code Base	14
Results after compilation	14
IOT Device Implementation	19
Wireframes	20
Mobile Application UI.....	Error! Bookmark not defined.
Gantt Chart	Error! Bookmark not defined.
Work Breakdown Structure.....	23

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.

The screenshot shows the Microsoft Teams calendar interface for September 2024. The sidebar includes links for Activity, Chat, Teams, Assignments, Calendar, Calls, OneDrive, and Apps. The main area displays the following meetings:

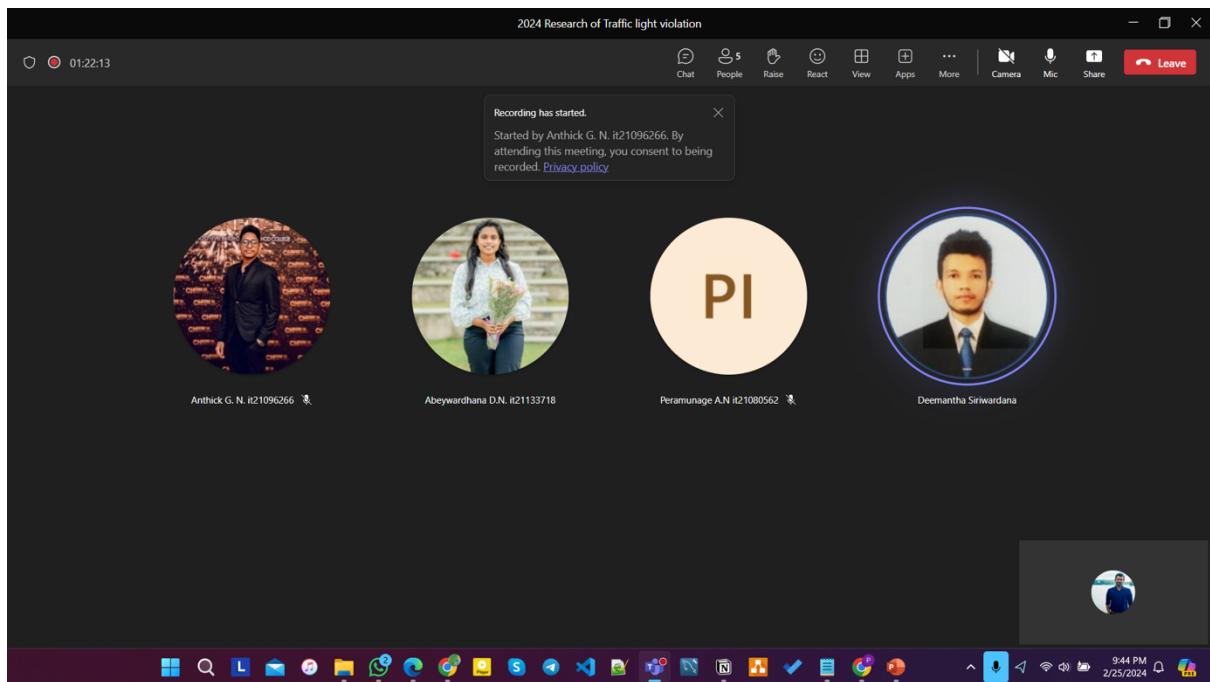
- 11 September | Today**: You don't have any meetings today. Schedule a meeting.
- 16 Mon**:
 - 22:00 - 1 hr: **Vehicle fire detection Research Group (Y4S1 2024)** (Microsoft Teams Meeting). Anthick G. N. it21096266 invited you, Abeywardhana D.N. it21133718 +3 others. Buttons: RSVP, Join, ...
- 17 Tue**:
 - 22:30 - 1 hr: **2024 Research of Traffic light violation** (Microsoft Teams Meeting). Anthick G. N. it21096266 invited you, Abeywardhana D.N. it21133718 +2 others. Buttons: RSVP, Join, ...
- 23 Mon**:
 - 22:00 - 1 hr: **Vehicle fire detection Research Group (Y4S1 2024)** (Microsoft Teams Meeting). Anthick G. N. it21096266 invited you, Abeywardhana D.N. it21133718 +3 others. Buttons: RSVP, Join, ...
- 24 Tue**:
 - 22:30 - 1 hr: **2024 Research of Traffic light violation** (Microsoft Teams Meeting). Anthick G. N. it21096266 invited you, Abeywardhana D.N. it21133718 +2 others. Buttons: RSVP, Join, ...
- 30 Mon**:
 - 22:00: **Vehicle fire detection Research Group (Y4S1 2024)** (Microsoft Teams Meeting). Buttons: RSVP, Join, ...

The screenshot shows a Microsoft Teams video call titled "Vehicle fire detection Research Group (Y4S1 2024)". The top bar includes controls for Chat, People (highlighted), Raise, React, View, Apps, More, Camera, Mic, Share, and Leave. The participants are:

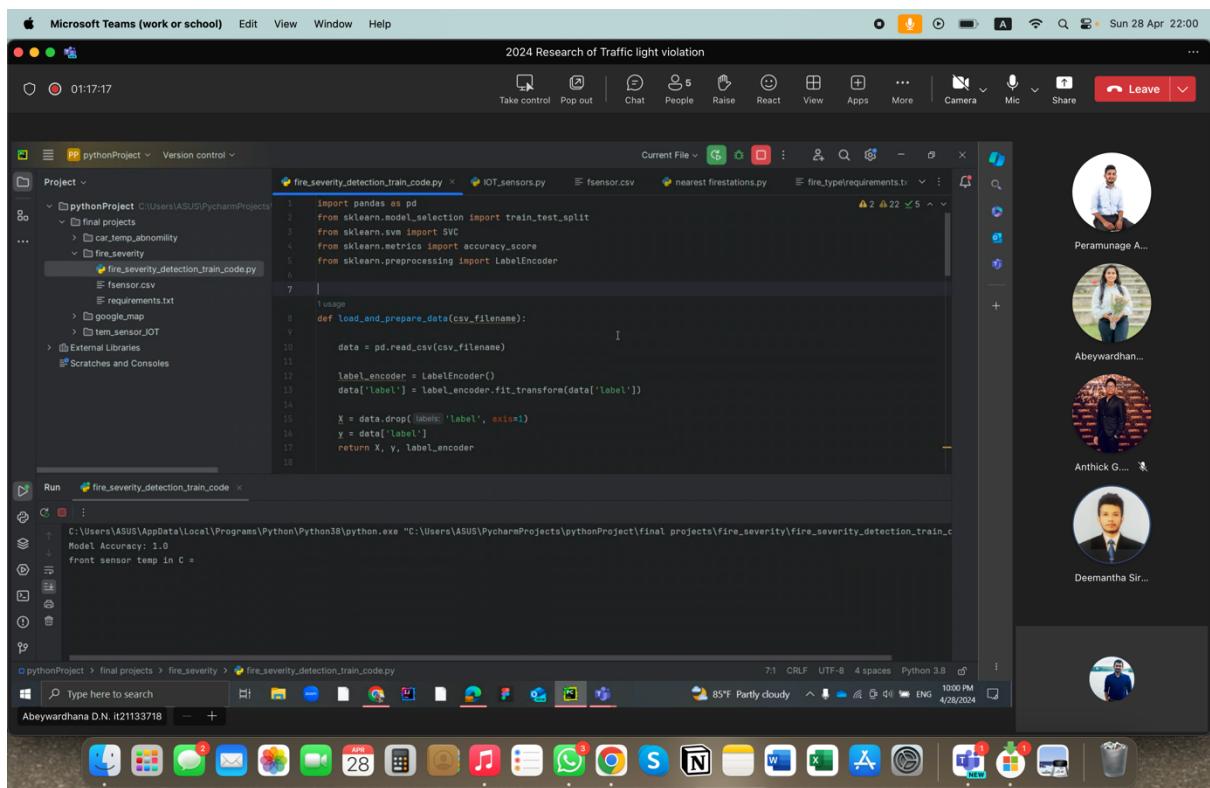
- Anthick G. N. it21096266 (Profile picture)
- Perumunage A.N it21080562 (Large circular placeholder for PI)
- Deemantha Siriwardana (Guest) (Large circular placeholder for DS)
- Abeywardhana D.N. it21133718 (Profile picture)
- Nelum Amarasinghe (Profile picture)
- Deemantha Siri... (Guest) (Placeholder for DS)
- Deemantha Siriwardana (Placeholder for DS)
- Others invited (1) (Placeholder for Deemantha Siriwardana)



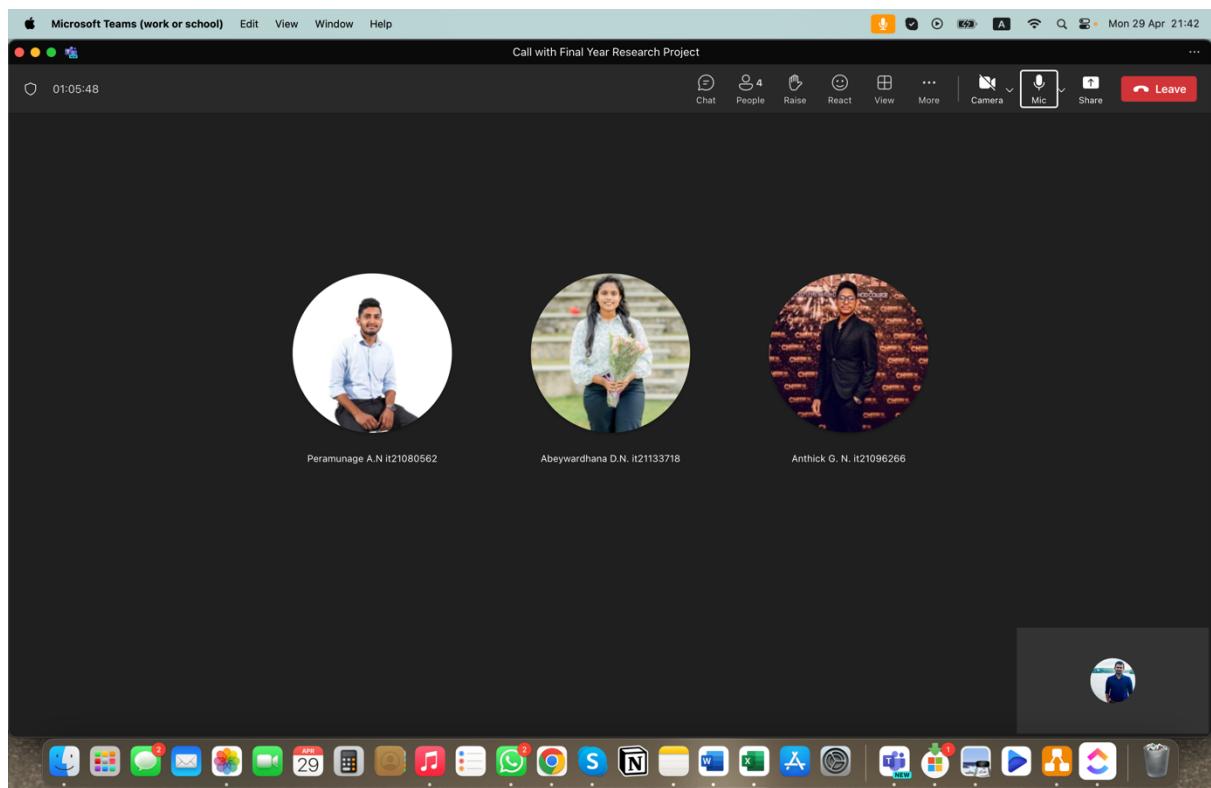
A detailed view of the Microsoft Teams meeting titled "2024 Research of Traffic light v...". The meeting was held on January 7, 2024, from 9:17 PM to 10:13 PM. A recording of the meeting is available, lasting 11m 53s. The recording details show the meeting was a "Co-Supervisor meeting" and was recorded by Abeywardhana D.N. It21133718. A message from Anthick G. N. is visible in the chat, asking if a meeting can be arranged within the next two days. The recording link is provided in the message.



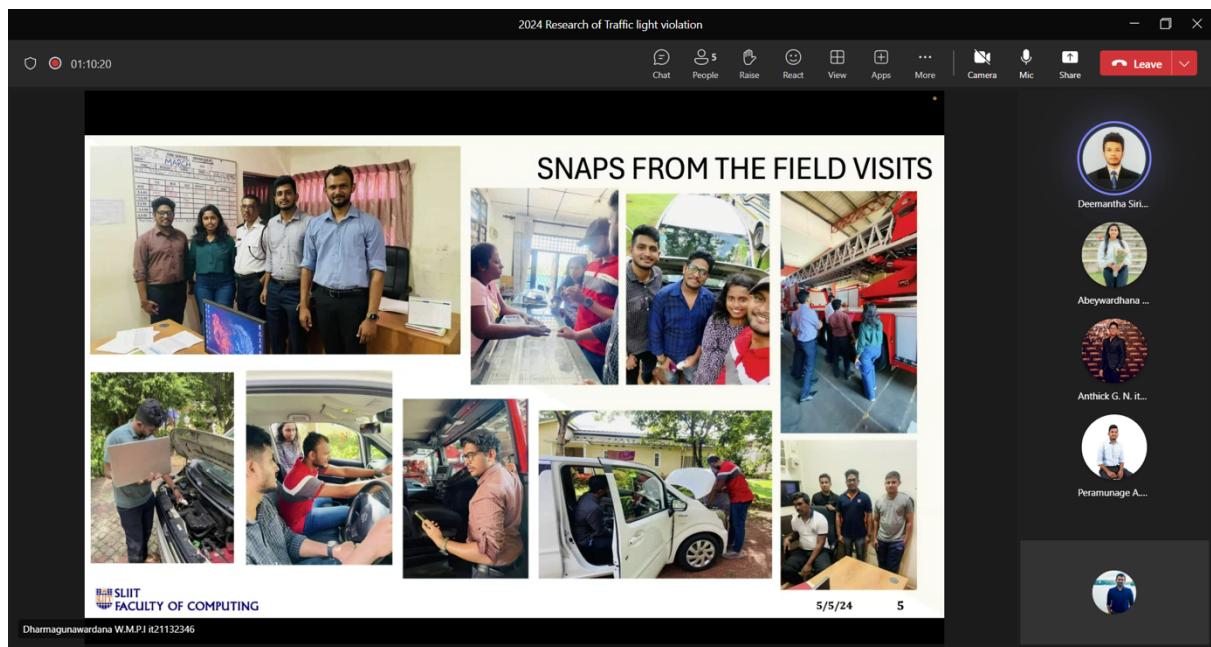
Code review with co-supervisor



Group meeting with group members



Presentation review with supervisor



Review Project Progress with Co-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

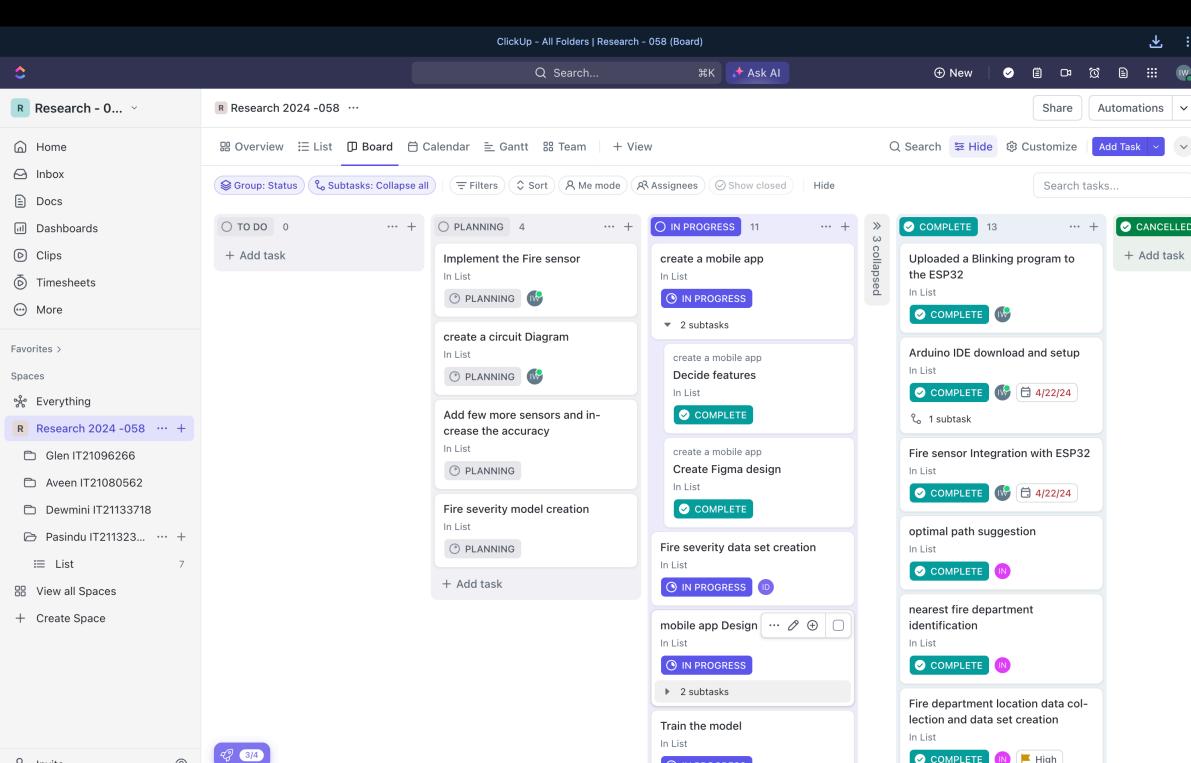
K.P.P.R Nanayakkara
Control Room officer

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

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

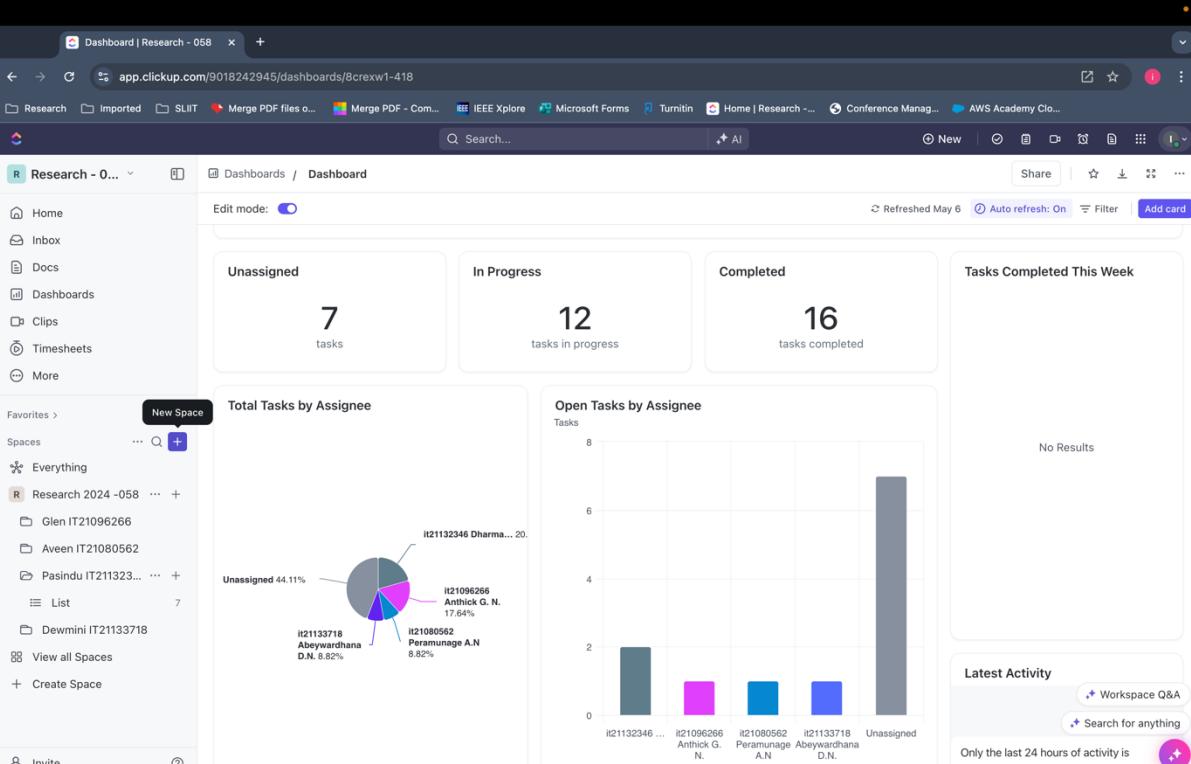
P.D.K.A.Wilson
Chief fire officer

Click Up Tasks Allocation



The screenshot shows a ClickUp tasks allocation board for the Research - 058 workspace. The board has four columns: TO DO (0 tasks), PLANNING (4 tasks), IN PROGRESS (11 tasks), and COMPLETE (13 tasks). The IN PROGRESS column contains several subtasks under the main task "create a mobile app". The COMPLETE column includes tasks like "Uploaded a Blinking program to the ESP32" and "Arduino IDE download and setup". The sidebar on the left shows the workspace navigation and a list of spaces.

Click Up Dashboard



The screenshot shows the ClickUp dashboard for the Research - 058 workspace. It features a summary section with counts for Unassigned (7), In Progress (12), Completed (16) tasks, and a section for Tasks Completed This Week. Below this are two data visualizations: a pie chart titled "Total Tasks by Assignee" showing the distribution of tasks among assignees, and a bar chart titled "Open Tasks by Assignee" showing the number of open tasks for each assignee. The sidebar on the left provides navigation and workspace management options.

Completed Tasks up to PP2

The screenshot shows a ClickUp dashboard titled 'Completed' with 16 tasks listed. The tasks are categorized under 'Tasks' and include:

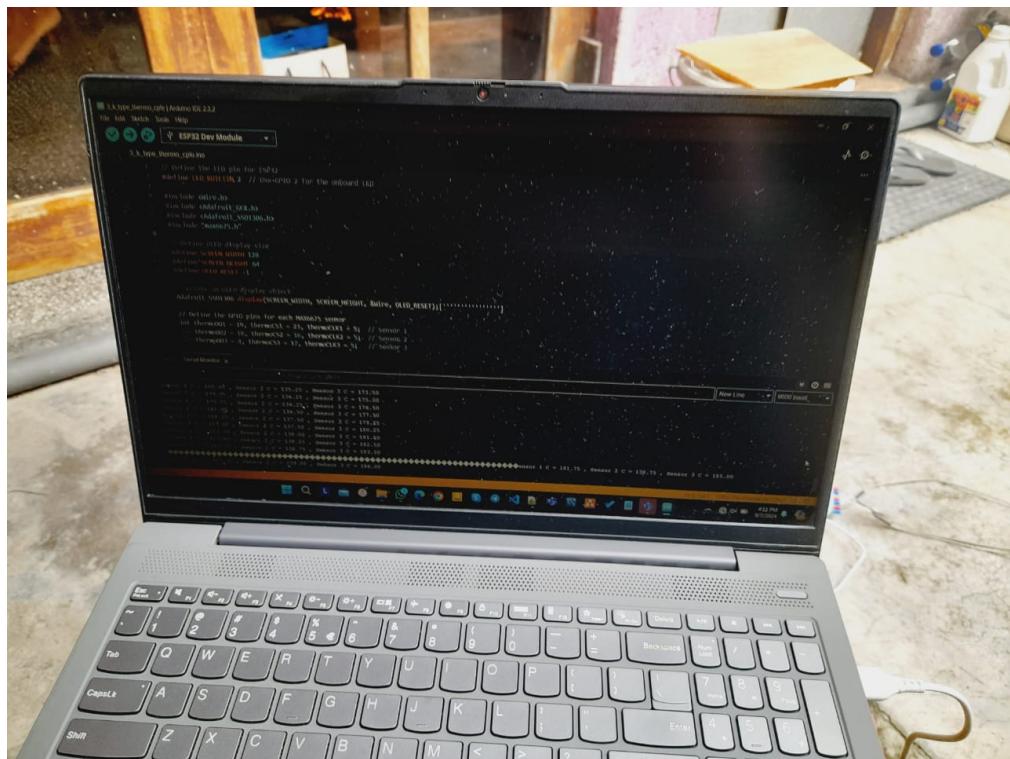
- Create Figma design
- Draw Swimmlane chart
- Add few more sensors and increase the accuracy
- Create a swimlane diagram
- fire department resource data collection
- nearest fire department identification
- optimal path suggestion
- Web application Wireframes Design (3 sub-tasks: Web Application Wireframes Design, Swimlane chart design, Web Application Wireframes Design)
- Fire department data collection

Each task has an assignee (represented by a small circular icon) and a status indicator (green circle with 'COM...' text). The ClickUp interface includes a search bar, a 'Customize' button, and a navigation bar at the top.

Project Implementation

Data Collection

Collecting temperature data by burning different types of materials, simulating deferent kind of Fires.





IOT Based Fire Extinguisher Mechanism

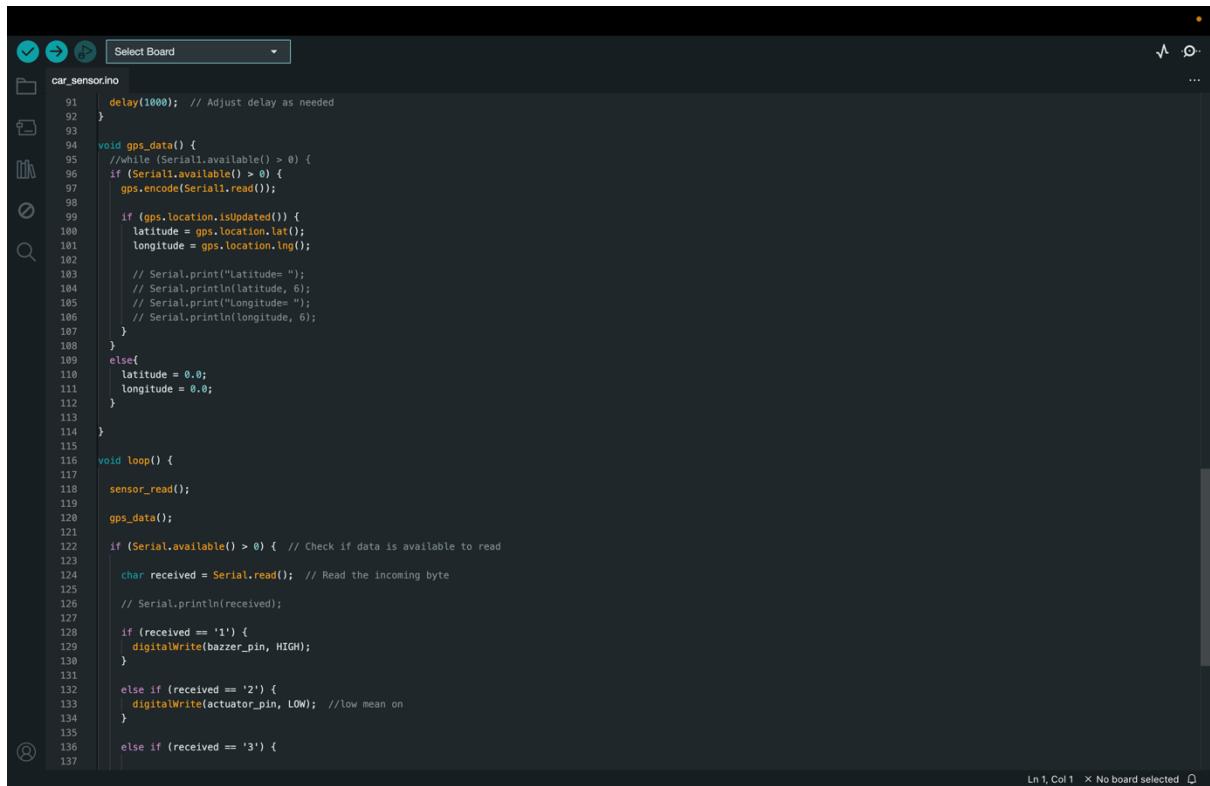
IOT Device Code Base

```
car_sensor.ino
1 #include <max6675.h>
2 #include <SoftwareSerial.h>
3 #include <TinyGPS++.h>
4
5 TinyGPSPlus gps;
6 SoftwareSerial ss(4, 3); // RX, TX
7
8 double latitude = 0.0;
9 double longitude = 0.0;
10
11 // Define Arduino SPI pins for each sensor
12 const int SCK_PIN_1 = 11; // Clock (SCK) pin for sensor 1
13 const int CS_PIN_1 = 12; // Chip Select (CS) pin for sensor 1
14 const int SO_PIN_1 = 13; // Serial Out (SO) pin for sensor 1
15 MAX6675 thermocouple1(SCK_PIN_1, CS_PIN_1, SO_PIN_1);
16
17 const int SCK_PIN_2 = 8; // Clock (SCK) pin for sensor 2
18 const int CS_PIN_2 = 9; // Chip Select (CS) pin for sensor 2
19 const int SO_PIN_2 = 10; // Serial Out (SO) pin for sensor 2
20 MAX6675 thermocouple2(SCK_PIN_2, CS_PIN_2, SO_PIN_2);
21
22 const int SCK_PIN_3 = 5; // Clock (SCK) pin for sensor 3
23 const int CS_PIN_3 = 6; // Chip Select (CS) pin for sensor 3
24 const int SO_PIN_3 = 7; // Serial Out (SO) pin for sensor 3
25 MAX6675 thermocouple3(SCK_PIN_3, CS_PIN_3, SO_PIN_3);
26
27 int buzzer_pin = 37;
28 int actuator_pin = 35;
29 int sensor1_pin = 33;
30
31
32 void setup() {
33
34
35     digitalWrite(buzzer_pin, LOW);
36     digitalWrite(actuator_pin, HIGH);
37
38     pinMode(buzzer_pin, OUTPUT); // Set pin 37 as an output
39     pinMode(actuator_pin, OUTPUT); // Set pin 35 as an output
40     pinMode(sensor1_pin, OUTPUT); // Set pin 33 as an output
41
42     digitalWrite(buzzer_pin, LOW);
43     digitalWrite(actuator_pin, HIGH);
44
45     delay(200);
46     Serial.begin(9600);
47 }
```

Ln 1, Col 1 × No board selected

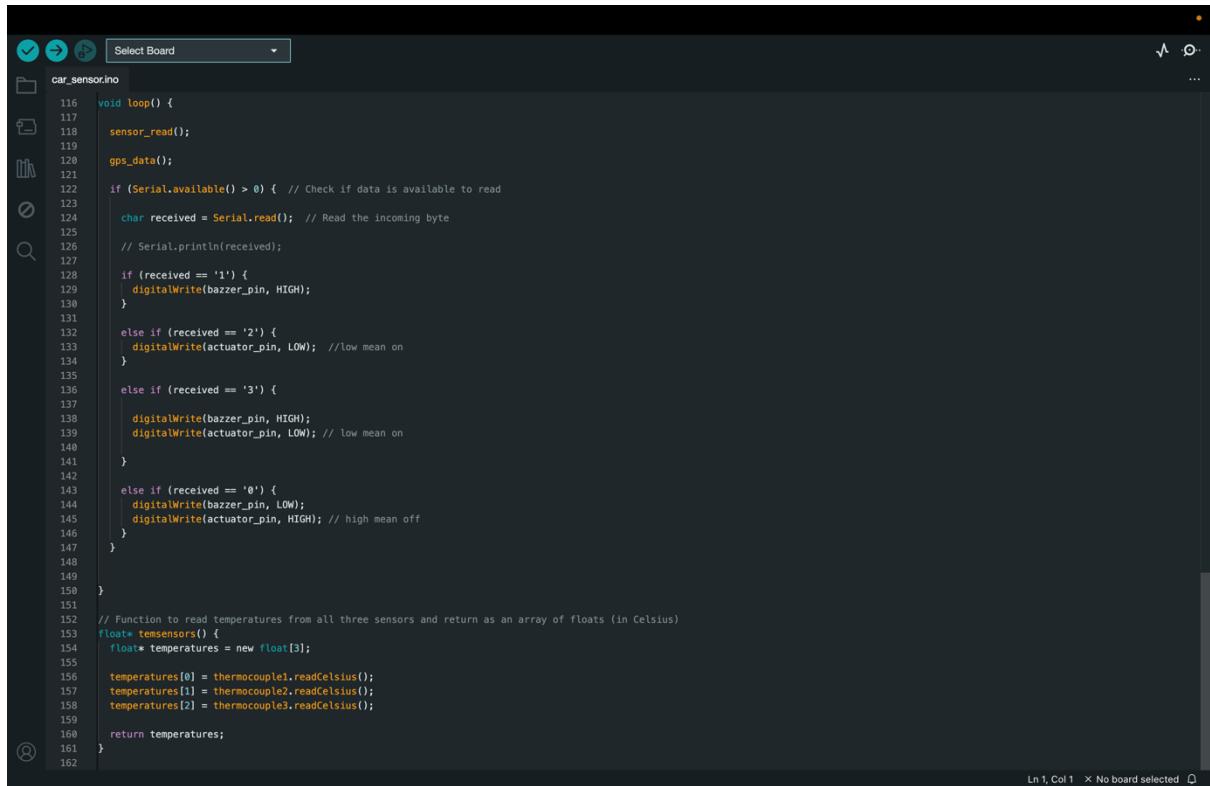
```
car_sensor.ino
45
46     delay(200);
47     Serial.begin(9600);
48     Serial.begin(9600);
49     delay(100);
50 }
51
52 void sensor_read() {
53
54     float* temps = temsensors();
55
56     // Print temperatures to Serial Monitor
57     //Serial.print("Sensor 1: ");
58     //Serial.print(int(temps[0]));
59     //Serial.print(",");
60
61     //Serial.print("Sensor 2: ");
62     //Serial.print(int(temps[1]));
63     //Serial.print(",");
64
65     //Serial.print("Sensor 3: ");
66     //Serial.print(int(temps[2]));
67     //Serial.print(",");
68     //Serial.println("\x00C");
69
70     Serial.print(int(1));
71     Serial.print(",");
72
73     Serial.print(int(0));
74     Serial.print(",");
75
76     Serial.print(int(1));
77     Serial.print(",");
78
79     Serial.print(latitude, 6);
80     Serial.print(",");
81
82     Serial.println(longitude, 6);
83     //Serial.print(",");
84
85     // Free memory allocated for temps array
86     delete[] temps;
87
88     delay(1000); // Adjust delay as needed
89
90 }
```

Ln 1, Col 1 × No board selected



```
car_sensor.ino
91     delay(1000); // Adjust delay as needed
92 }
93
94 void gps_data() {
95     //While (Serial.available() > 0) {
96     if (Serial1.available() > 0) {
97         gps.encode(Serial1.read());
98
99         if (gps.location.isUpdated()) {
100             latitude = gps.location.lat();
101             longitude = gps.location.lng();
102
103             // Serial.print("Latitude= ");
104             // Serial.println(latitude, 6);
105             // Serial.print("Longitudes= ");
106             // Serial.println(longitude, 6);
107         }
108     } else{
109         latitude = 0.0;
110         longitude = 0.0;
111     }
112 }
113
114 }
115
116 void loop() {
117     sensor_read();
118     gps_data();
119
120     if (Serial.available() > 0) { // Check if data is available to read
121
122         char received = Serial.read(); // Read the incoming byte
123
124         // Serial.println(received);
125
126         if (received == '1') {
127             digitalWrite(buzzer_pin, HIGH);
128         }
129
130         else if (received == '2') {
131             digitalWrite(actuator_pin, LOW); //low mean on
132         }
133
134         else if (received == '3') {
135
136             digitalWrite(buzzer_pin, HIGH);
137             digitalWrite(actuator_pin, LOW); // low mean on
138
139         }
140
141         else if (received == '0') {
142             digitalWrite(buzzer_pin, LOW);
143             digitalWrite(actuator_pin, HIGH); // high mean off
144         }
145
146     }
147 }
148
149
150 }
151
152 // Function to read temperatures from all three sensors and return as an array of floats (in Celsius)
153 float* temsensors() {
154     float* temperatures = new float[3];
155
156     temperatures[0] = thermocouple1.readCelsius();
157     temperatures[1] = thermocouple2.readCelsius();
158     temperatures[2] = thermocouple3.readCelsius();
159
160     return temperatures;
161 }
162
```

Ln 1, Col 1 × No board selected



```
car_sensor.ino
116 void loop() {
117     sensor_read();
118     gps_data();
119
120     if (Serial.available() > 0) { // Check if data is available to read
121
122         char received = Serial.read(); // Read the incoming byte
123
124         // Serial.println(received);
125
126         if (received == '1') {
127             digitalWrite(buzzer_pin, HIGH);
128         }
129
130         else if (received == '2') {
131             digitalWrite(actuator_pin, LOW); //low mean on
132         }
133
134         else if (received == '3') {
135
136             digitalWrite(buzzer_pin, HIGH);
137             digitalWrite(actuator_pin, LOW); // low mean on
138
139         }
140
141         else if (received == '0') {
142             digitalWrite(buzzer_pin, LOW);
143             digitalWrite(actuator_pin, HIGH); // high mean off
144         }
145
146     }
147 }
148
149
150 }
151
152 // Function to read temperatures from all three sensors and return as an array of floats (in Celsius)
153 float* temsensors() {
154     float* temperatures = new float[3];
155
156     temperatures[0] = thermocouple1.readCelsius();
157     temperatures[1] = thermocouple2.readCelsius();
158     temperatures[2] = thermocouple3.readCelsius();
159
160     return temperatures;
161 }
162
```

Ln 1, Col 1 × No board selected

Results after compilation

The screenshot shows the Arduino IDE interface with the sketch name "3_k_type_thermo_couple". The "Serial Monitor" tab is active, displaying a continuous stream of sensor data. The data consists of three lines of text, each containing three sensor readings: Sensor 1 C, Sensor 2 C, and Sensor 3 C. The values fluctuate slightly between 28.00 and 29.50 degrees Celsius.

```

Sensor 1 C = 29.25 , Sensor 2 C = 28.50 , Sensor 3 C = 29.00
Sensor 1 C = 28.75 , Sensor 2 C = 28.25 , Sensor 3 C = 29.25
Sensor 1 C = 28.75 , Sensor 2 C = 28.25 , Sensor 3 C = 29.00
Sensor 1 C = 29.00 , Sensor 2 C = 28.50 , Sensor 3 C = 29.25
Sensor 1 C = 29.25 , Sensor 2 C = 28.50 , Sensor 3 C = 29.25
Sensor 1 C = 29.00 , Sensor 2 C = 28.75 , Sensor 3 C = 29.25
Sensor 1 C = 29.25 , Sensor 2 C = 28.75 , Sensor 3 C = 29.50
Sensor 1 C = 29.50 , Sensor 2 C = 28.50 , Sensor 3 C = 28.75
Sensor 1 C = 29.00 , Sensor 2 C = 29.00 , Sensor 3 C = 29.75
Sensor 1 C = 29.00 , Sensor 2 C = 28.50 , Sensor 3 C = 29.25
Sensor 1 C = 29.25 , Sensor 2 C = 28.75 , Sensor 3 C = 29.00
Sensor 1 C = 29.50 , Sensor 2 C = 28.50 , Sensor 3 C = 29.50
Sensor 1 C = 28.75 , Sensor 2 C = 28.75 , Sensor 3 C = 29.50
Sensor 1 C = 29.00 , Sensor 2 C = 28.50 , Sensor 3 C = 29.25
Sensor 1 C = 29.00 , Sensor 2 C = 28.25 , Sensor 3 C = 29.00
Sensor 1 C = 29.25 , Sensor 2 C = 28.50 , Sensor 3 C = 29.25

```

Main File – which connects Sensor data with ML models

The screenshot shows the PyCharm IDE interface with the project "FINAL PROJECT COMPLIT..." open. The "main.py" file is the active editor. The code in the main.py file is as follows:

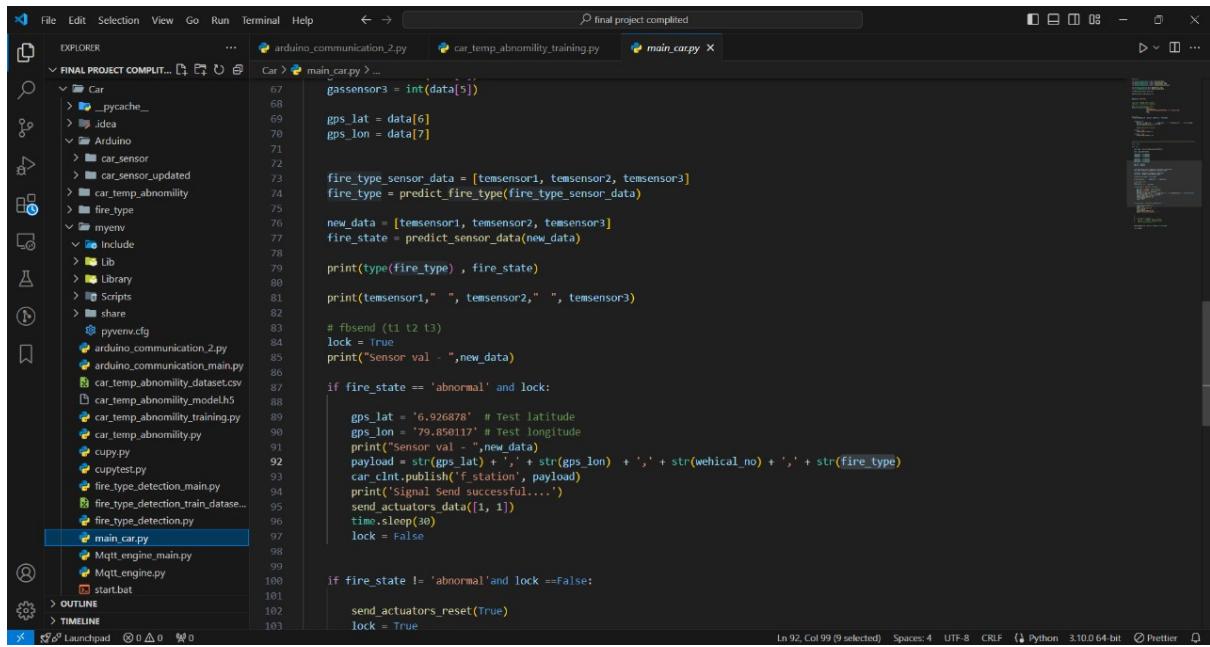
```

# Demo mode
def demo_mood(gps_lat , gps_lon , vehical_no , fire_type):
    if msg == 'fire':
        payload = str(gps_lat) + ',' + str(gps_lon) + ',' + str(vehical_no)+ ',' + str(fire_type)
        car_cln.publish('f',payload)
        # send_actuators_data([1, 1])
        print('Signal Send successful....')
    if msg == 'reset':
        send_actuators_reset(True)
    if msg == 'ftype':
        send_actuators_data([1, 1])

# main while loop*****
while True:
    msg, topic = car_cln.subscribe_last_buffer()
    data = get_sensor_data()

    temsensor1 = int(data[0])
    temsensor2 = int(data[1])
    temsensor3 = int(data[2])
    gassensor1 = int(data[3])
    gassensor2 = int(data[4])

```

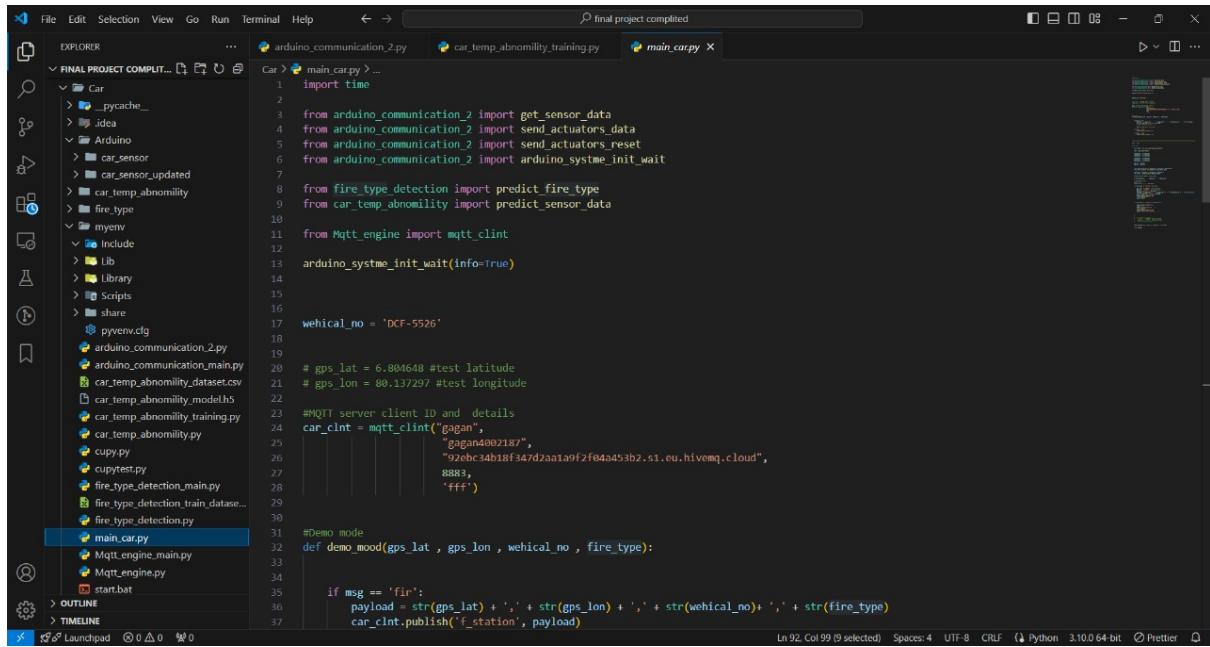


```

File Edit Selection View Go Run Terminal Help ← → final project compiled
EXPLORER Car > main_car.py ...
FINAL PROJECT COMPLIT... _pycache_ idea Arduino car_sensor car_sensor_updated car_temp_abnomility fire_type myenv Lib Library Scripts share pyvenv.cfg arduino_communication_2.py arduino_communication_main.py car_temp_abnomility_dataset.csv car_temp_abnomility_model.h5 car_temp_abnomility_training.py cupy.py cupytest.py fire_type_detection_main.py fire_type_detection_train_database.py fire_type_detection.py main_car.py Mqtt_engine_main.py MqttEngine.py start.bat
OUTLINE TIMELINE
Launchpad 0 0 0
main_car.py ...
67     gassensor3 = int(data[5])
68
69     gps_lat = data[6]
70     gps_lon = data[7]
71
72     fire_type_sensor_data = [temsensor1, temsensor2, temsensor3]
73     fire_type = predict_fire_type(fire_type_sensor_data)
74
75     new_data = [temsensor1, temsensor2, temsensor3]
76     fire_state = predict_sensor_data(new_data)
77
78     print(type(fire_type), fire_state)
79
80     print(temsensor1, " ", temsensor2, " ", temsensor3)
81
82     # fbsend (t1 t2 t3)
83     lock = True
84
85     print("Sensor val - ", new_data)
86
87     if fire_state == 'abnormal' and lock:
88
89         gps_lat = '6.926878' # Test latitude
90         gps_lon = '79.850117' # Test longitude
91         print("Sensor val - ", new_data)
92         payload = str(gps_lat) + ',' + str(gps_lon) + ',' + str(vehical_no) + ',' + str(fire_type)
93         car_cint.publish('f_station', payload)
94         print("Signal Send successful....")
95         send_actuators_data([1, 1])
96         time.sleep(30)
97         lock = False
98
99     if fire_state != 'abnormal' and lock == False:
100
101         send_actuators_reset(True)
102         lock = True
103

```

Ln 92, Col 99 (9 selected) Spaces: 4 UTF-8 CRLF ⓘ Python 3.10.0 64-bit ⓘ Prettier



```

File Edit Selection View Go Run Terminal Help ← → final project compiled
EXPLORER Car > main_car.py ...
FINAL PROJECT COMPLIT... _pycache_ idea Arduino car_sensor car_sensor_updated car_temp_abnomility fire_type myenv Lib Library Scripts share pyvenv.cfg arduino_communication_2.py arduino_communication_main.py car_temp_abnomility_dataset.csv car_temp_abnomility_model.h5 car_temp_abnomility_training.py car_temp_abnomility.py cupy.py cupytest.py fire_type_detection_main.py fire_type_detection_train_database.py fire_type_detection.py main_car.py Mqtt_engine_main.py MqttEngine.py start.bat
OUTLINE TIMELINE
Launchpad 0 0 0
main_car.py ...
1 import time
2
3 from arduino_communication_2 import get_sensor_data
4 from arduino_communication_2 import send_actuators_data
5 from arduino_communication_2 import send_actuators_reset
6 from arduino_communication_2 import arduino_system_init_wait
7
8 from fire_type_detection import predict_fire_type
9 from car_temp_abnomility import predict_sensor_data
10
11 from Mqtt_engine import mqtt_cint
12
13 arduino_system_init_wait(info=True)
14
15
16 wehical_no = 'DCF-5526'
17
18
19 # gps lat = 6.804648 #test latitude
20 # gps lon = 80.137297 #test longitude
21
22
23 #MQTT server client ID and details
24 car_cint = mqtt_cint("gagan",
25                         "gagan002187",
26                         "92ebc34bf18f347d2aa1a9f2f04a453b2.s1.eu.hivemq.cloud",
27                         8883,
28                         "ffff")
29
30
31 #Demo mode
32 def demo_mood(gps_lat , gps_lon , wehical_no , fire_type):
33
34
35     if msg == 'fir':
36         payload = str(gps_lat) + ',' + str(gps_lon) + ',' + str(wehical_no)+ ',' + str(fire_type)
37         car_cint.publish('f_station', payload)

```

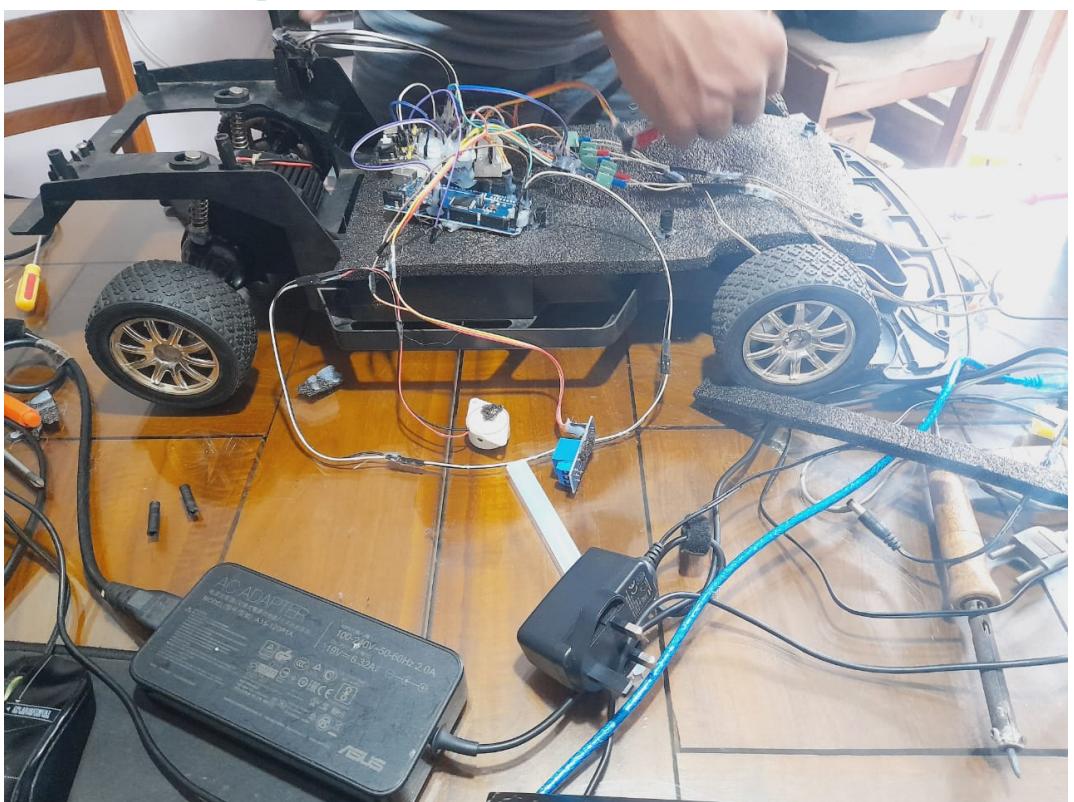
Ln 92, Col 99 (9 selected) Spaces: 4 UTF-8 CRLF ⓘ Python 3.10.0 64-bit ⓘ Prettier

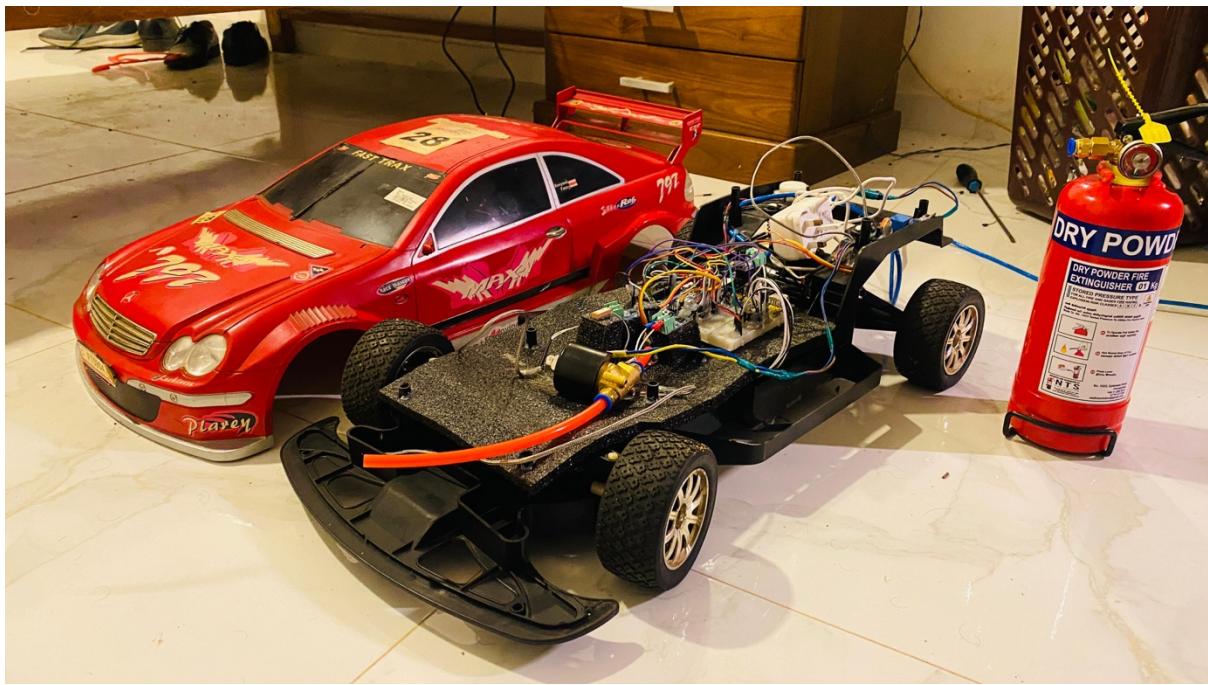
After Running the main program to collect the sensor data.

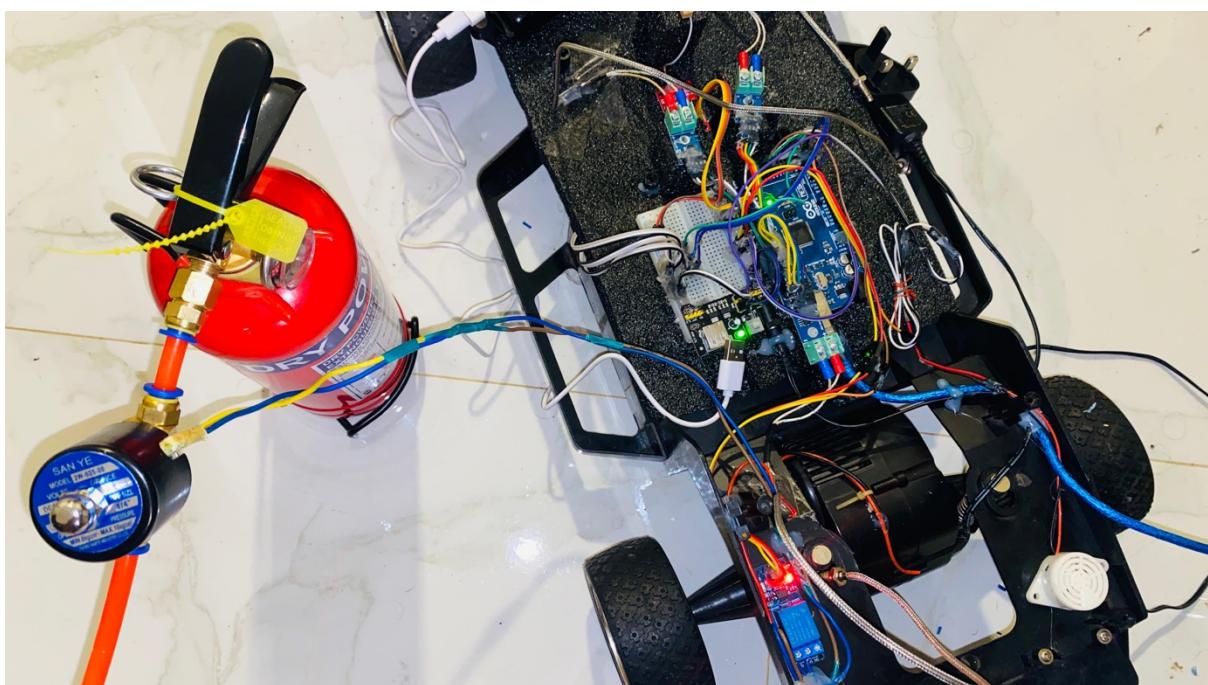
The screenshot shows a Python development environment with the following details:

- Project Explorer:** A tree view showing the project structure:
 - FINAL PROJECT COMPLETED
 - Car
 - __pycache__
 - idea
 - Arduino
 - car_temp_abnormality
 - fire_type
 - mynav
 - Include
 - lib
 - library
 - Scripts
 - share
 - pyvenv.cfg
 - arduino_communication_2.py
 - arduino_communication_main.py
 - car_temp_abnormality_dataset.csv
 - car_temp_abnormality_model.h5
 - car_temp_abnormality_training.py
 - car_temp_abnormality.py
 - cupy.py
 - cupytest.py
 - fire_type_detection_main.py
 - fire_type_detection_train_dataset...
 - fire_type_detection.py
 - main_car.py
 - Mqtt_engine_main.py
 - Mqtt_engine.py
 - start.bat
- Code Editor:** The main editor window displays the content of `main_car.py`. The code handles GPS data, vehicle number, fire type, and actuator control. It includes a `demo_mood` function and a `time.sleep(1)` call.
- Terminal:** The terminal tab shows command-line output for several files, each with a duration of 20ms/step.
- Bottom Bar:** Includes icons for launchpad, outline, timeline, and other development tools.

IOT Device Implementation









Work Breakdown Structure

