Project IOT

Luminus

Team Marie²

Marie GUMB
Marie-Léonie SERIZOT

Team Marie²

Marie

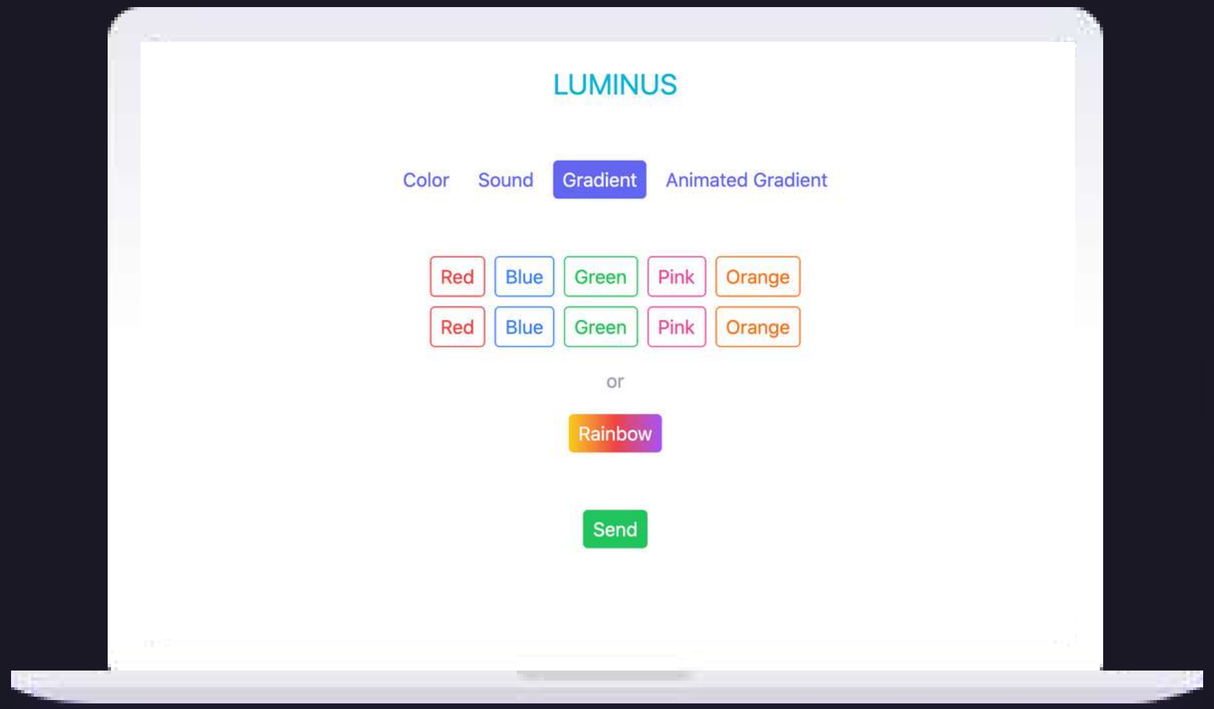
- Conception, Front and Back

Marie-Léonie

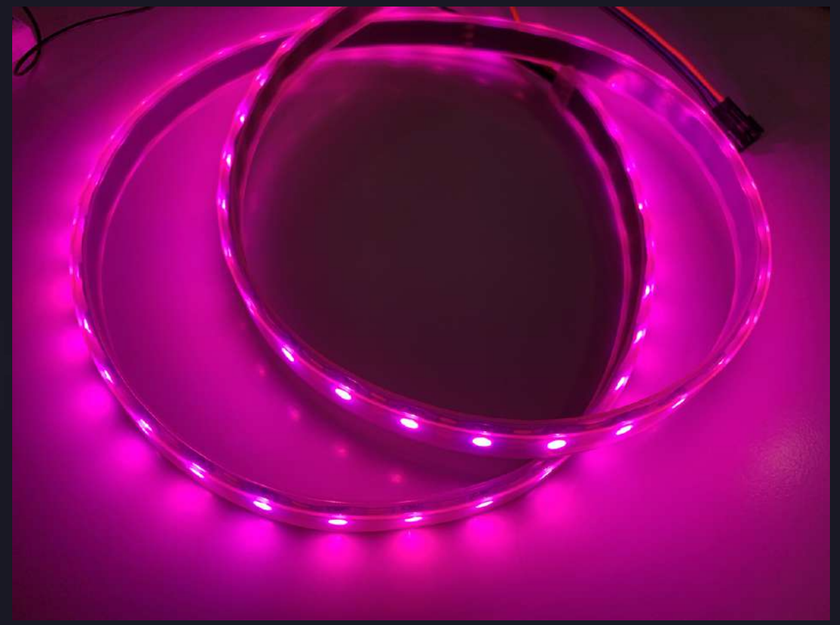
- Hardware, Conception, Assembly

What is our project?

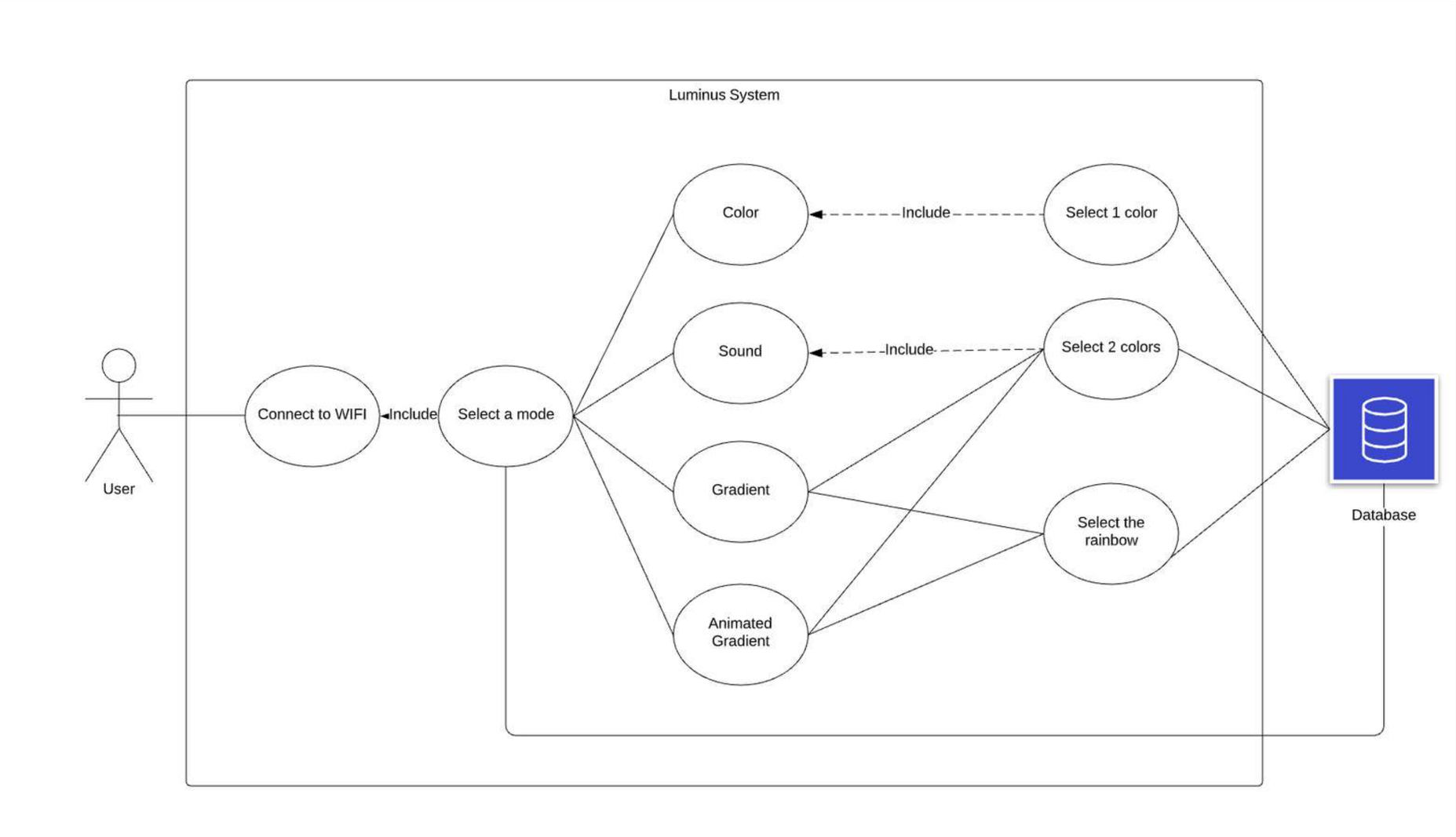
- Create a programs for RGB LED : Color | Sound | Gradient | Animated Gradient.
- Let the user choose light programs on our interface.



+



Use Case


Luminus System
Color
Include
Select 1 color
Sound
Include
Select 2 colors
Gradient
Select the rainbow
Animated Gradient
Connect to WIFI
Include
Select a mode
User
Database

State of arts

Philips Hue :

The philips hue led strip is directly connected to the HDMI stream, which allows it to react according to the colors on the screen.

Challenge & motivation

Motivation

- Connected leds for our own home : when we're partying, or just to create an atmosphere
- Create something that everyone could use and enjoy

Challenge

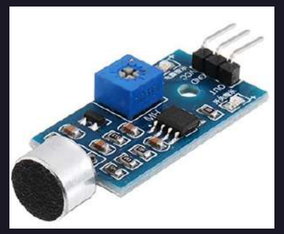
- Create a usefulinterface
- Use a database
- Code in arduino a lot of methods

Components

- ESP32 (1)

- RGB Led Neopixel (1)

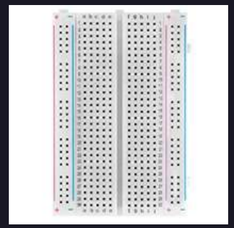
- Sound Sensor (1)



- Screen OLED ssd1306 (1)

- Wires (12)

- Breadboard (1)



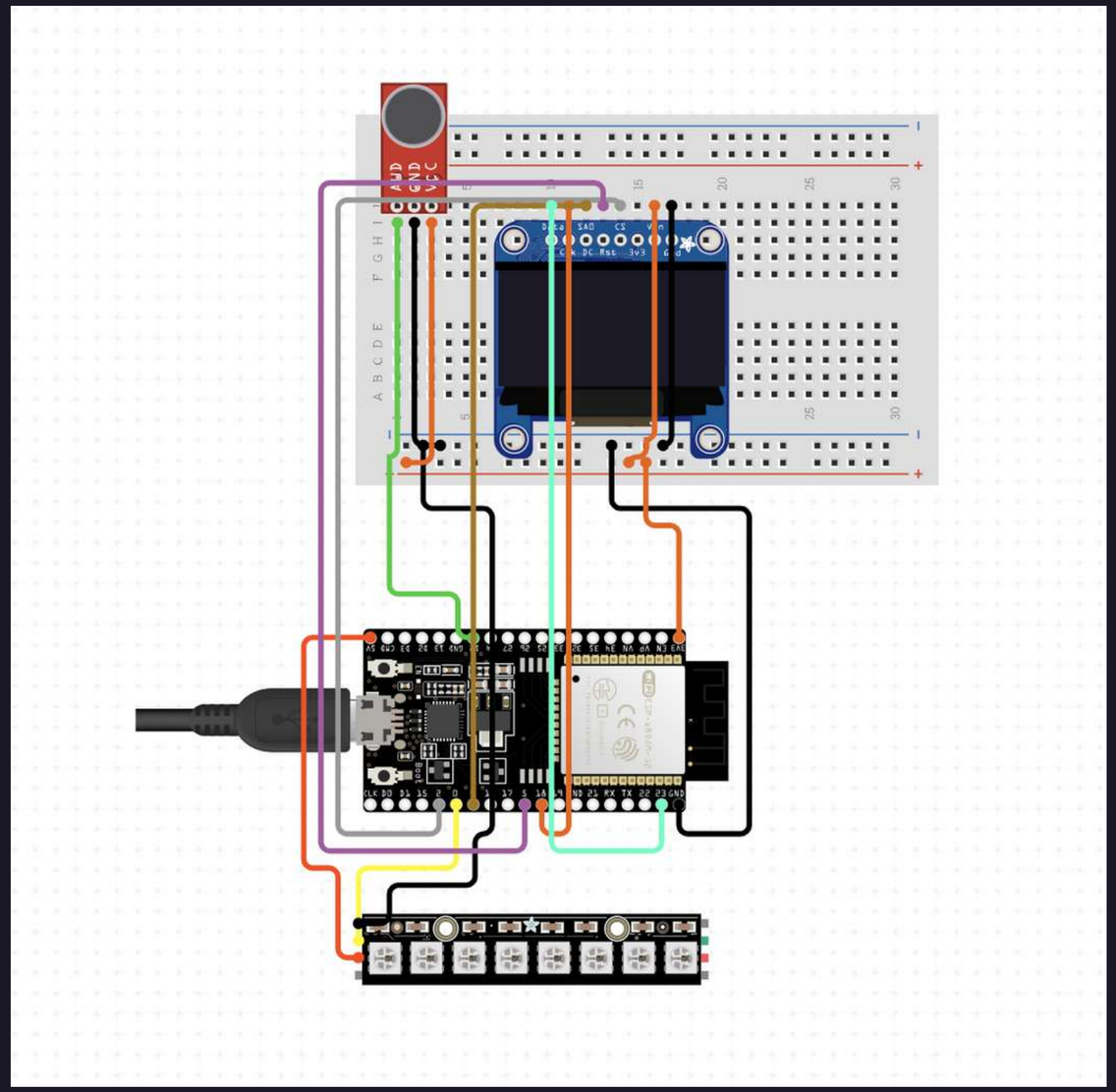
Structure

Firebase

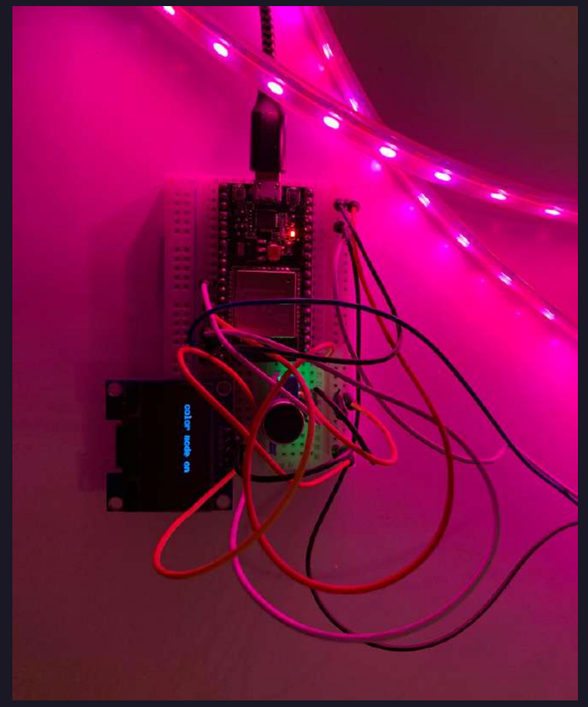
HTTP request
(GET)

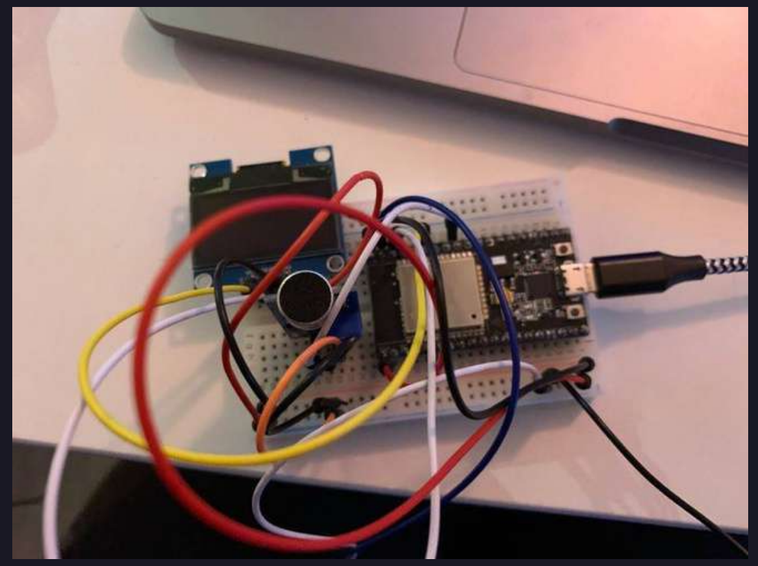
HTTP request
(POST)

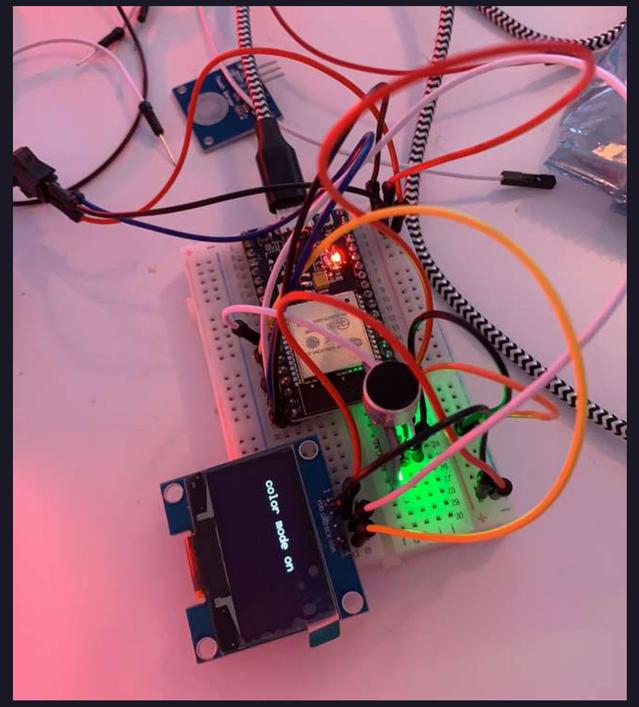
Connection



Connection







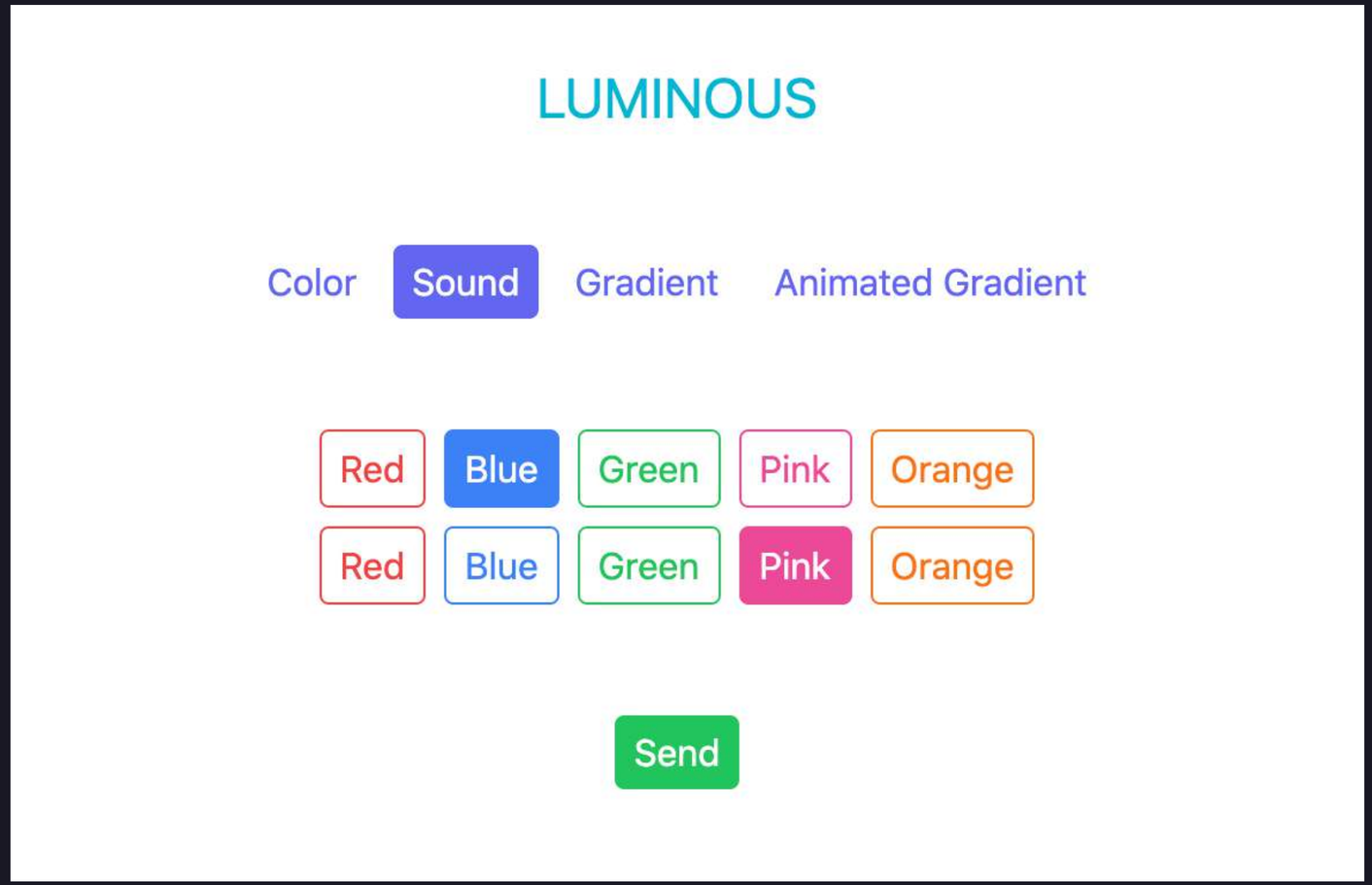
User interface

We use :

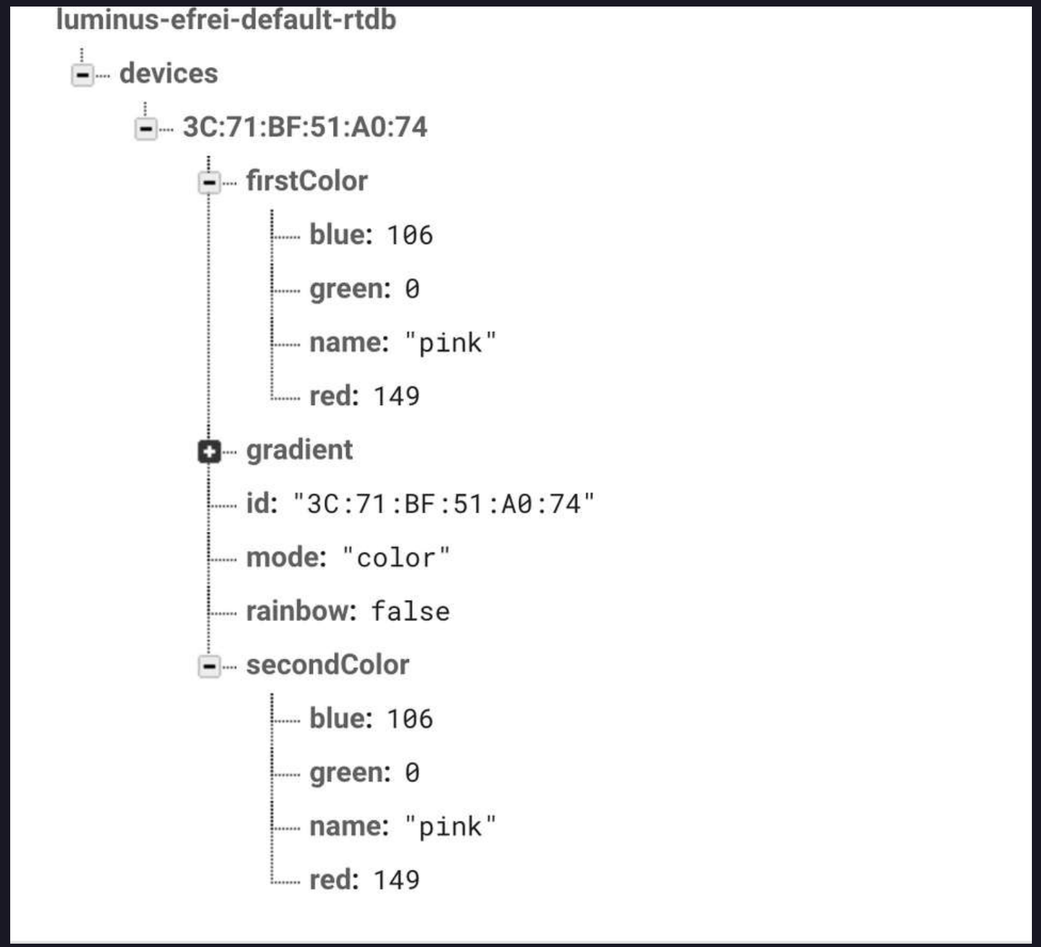
- VueJs
- Tailwind
- Realtime Database of firebase

The user can choose :

- Mode
- Colors



The Database



Code

Some library :

Adafruit_Neopixel (led)
Json
HttpClient (request http)
Adafruit_SH1106 (oled screen)

```
void loop(){
     if (millis() - previousTime >= 5*1000UL){ // toute les 5s
       previousTime = millis();
       fetchDeviceState();
     }

    String selectedMode = JSON.stringify(deviceState["mode"]);
    selectedMode.replace("\"", "");

    if(selectedMode.equals("color")){
        displayLog("color mode on");
        setColor();
        delay(50);
    } else if(selectedMode.equals("sound")){
        displayLog("sound mode on");
        soundMode();
        delay(50);
    } else if(selectedMode.equals("gradient")){
        displayLog("gradient mode on");
        gradientMode();
        delay(50);
    } else if(selectedMode.equals("animated-gradient")){
        displayLog("animated gradient mode on");
        animatedGradientMode();
    } else {
        displayLog("no mode recognized");
    }
}
```

Code

```
void animatedGradientMode(){
    JSONVar color = deviceState["gradient"];

    Serial.println(color[0]);

    for(int i = 0; i < NUMPIXELS; i++) {
        int r = (int) color[i][0];
        int g = (int) color[i][1];
        int b = (int) color[i][2];
        int numPixel = (i + decalage) % NUMPIXELS;
        pixels.setPixelColor(numPixel, pixels.Color(r, g, b));
    }
    decalage ++;
    pixels.show();
}
```

```
void gradientMode(){
    JSONVar color = deviceState["gradient"];

    Serial.println(color[0]);

    for(int i = 0; i < NUMPIXELS; i++) {
        int r = (int) color[i][0];
        int g = (int) color[i][1];
        int b = (int) color[i][2];
        pixels.setPixelColor(i, pixels.Color(r, g, b));
        delay(10);
    }
    decalage ++;
    pixels.show();
}
```

Difficulties

- Initially we wanted to make another mode using a light sensor,

but we couldn't solder it and therefore use it.

- Coding in arduino limits the possibilities for database queries:

with python we could connect in real time to the database and

not request every x seconds.

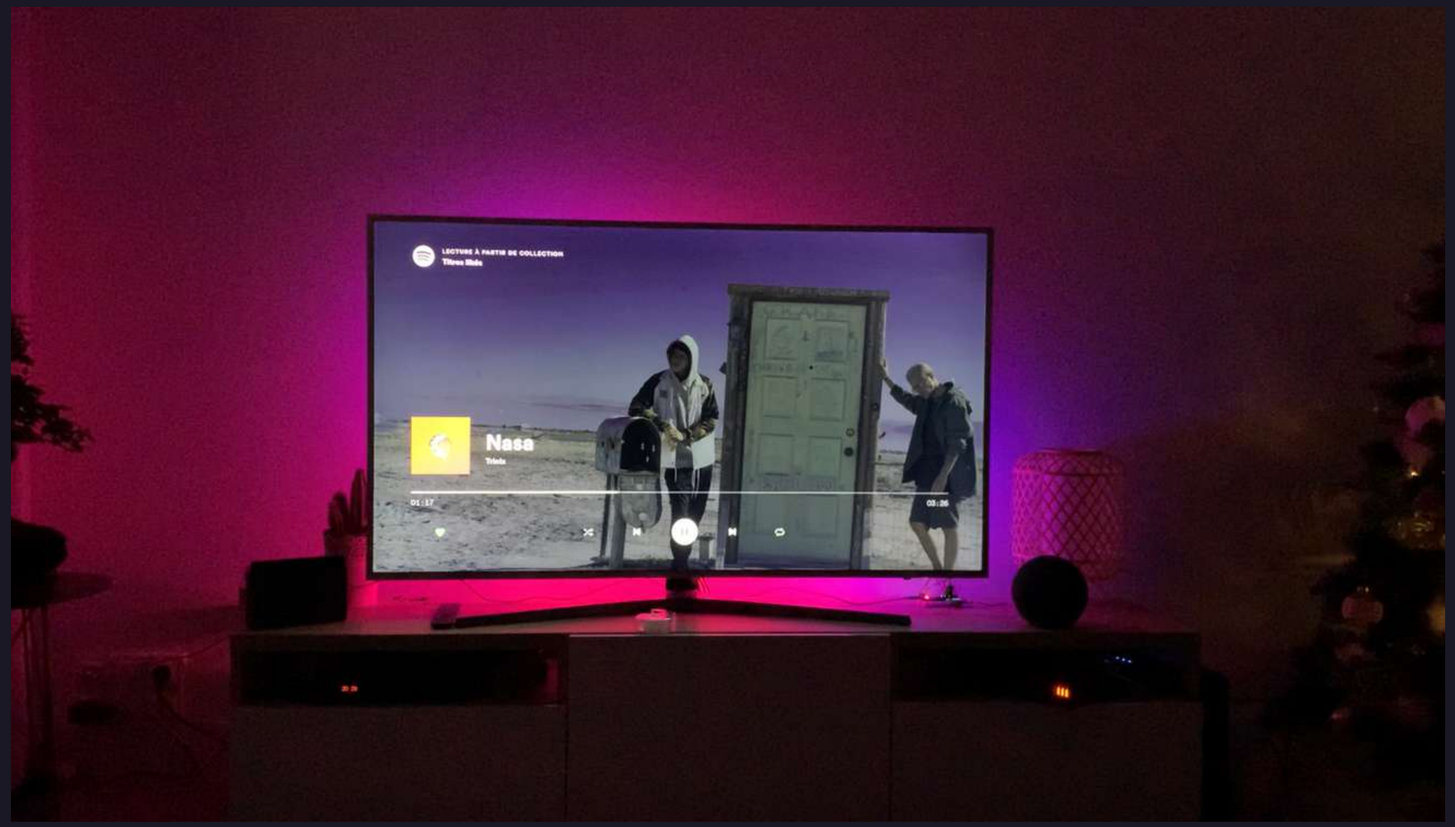
What can be improved

- A better sound sensor to better reacting music (with bass and treble or frequencies)

- Create a case to hide hardware

- Add more functionalities

THANK YOU


LECTURE À PARTIR DE COLLECTION
Titres Rnbs
Nasa
Travis
01:17
03:36