VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Smart Exhaustion and Evacuation System

Final Report

Team Members:
Sewanti Lahiri 20BEE0007
Siddhanta Mondal 20BEE0010

Course Name: Internet of Things Fundamentals
Course Code: ECE3501

Department: School of Electrical and Electronics Engineering
Date: November 15, 2022

1 ACKNOWLEDGEMENT

In successfully bringing this project to it's final form, we have come across many people whose contributions have helped us in many ways. We would like to thank all those who are related to this project. Primarily, we would like to express our gratitude to Vijaya Priya Ma'am, whose valuable guidance, monitoring and suggestions at each and every stage of our work has helped us with the successful completion of our project. We will always be thankful to her in this regard. We consider ourselves very fortunate for being able to work with a very considerate and encouraging professor like her. Her involvement with the originality has triggered and nourished our intellectual maturity that will help us for a long time to come. As one of the team members I would also like to appreciate my team mate for her coordination, concern and support. Without her, our project would not have been a success. We would also like to thank our parents who helped us a lot in finishing our project by helping us buying the facilities that were required for our project and also encouraged us to do it with a positive attitude.

2 ABSTRACT

In industries and workplaces, the safety of the workers is the topmost concern of any owner. In situations of accidental fire within factories and industries, the proper evacuation of the workers and the proper exhaustion of the smoke can protect the lives of the workers and also reduce the damage done to the factory. With the implementation of IoT, we can automate several operations like, automatic smoke sensing, automatic door opening, automatic initialisation of sprinkler system, notifying the owner or the safety officer immediately in times of need without any need for human intervention, etc.
This study examines the various utilities and functions, one can develop with the help of Internet of Things. Along with that, the components used in the project are much cheaper than the components available in the market currently, thus providing an economic advantage to the customers.
The suggested solutions will be able to alleviate the issues that arise as a result of the lack of a dependable, efficient and modern smoke detection system while the economic analysis technique will aid in the feasibility study of the projects.

3 INTRODUCTION

In this project we are using Arduino UNO as the parent microcontroller and ESP8266 as a WiFi module. We have interfaced both these devices together with the help of I2C serial communication protocol. Thus, data can be seamlessly transferred between these two devices. We have connected a smoke sensor (MQ-2) with the Arduino which will continuously monitor the smoke level. In case, the smoke levels crosses the normal threshold value, Arduino will initiate a series of events, that will enable the workers to move to safety, as well as, take actions in minimising the smoke levels and also notifying the owner or safety officier via SMS on phone.

For easier understanding and execution of the project, we are diving the project into 3 parts:

Part 1

In this part, we are setting up the Arduino UNO with the smoke sensor. Along with that, we have connected 2 LEDs to indicate the status of the room. We have also connected a buzzer, that will start once the smoke level crosses the threshold value. We also have conncted a exhaust fan, that will help in reducing the smoke level. Along with this, there is also a provision of a automatic sprinkler system, that will work to put out any fire that might have caused the smoke.
Once, the circuit had been made, we tested it by putting it in several conditions to test it's reliability and efficiency.

Part 2

In this section, we have worked on setting up a SMS system, that will alert the safety officer immediately in times of accident, without any need for any human intervention. For this, we have used the IFTTT website. We used webhooks and Ardroid SMS feature of the website to send an SMS. This service uses the HTTPs protocol to send message with the help of internet.

Part 3

In this part, we set up the NodeMCU for sending the SMS to the owner's phone. For that, we first connected the NodeMCU with the WiFi or personal hotspot. After that, we interfaced the Arduino with NodeMCU for sending and receiving data serially between them.
After that, we connected the IFTTT platform with it. As soon as, the smoke level exceed the threshold, it will trigger a web request to the website and that will inturn send an SMS to phone.

4 BLOCK DIAGRAM

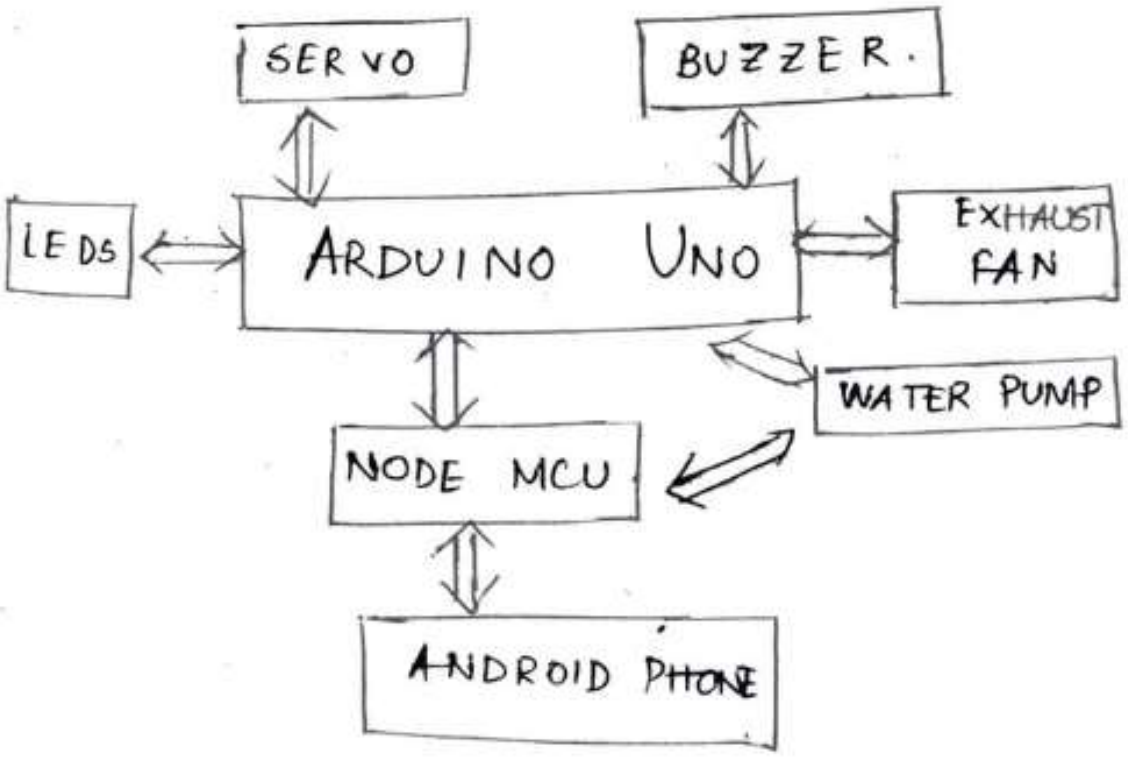


Figure 1: Block Diagram

5 EXPLANATION OF THE BLOCK DIAGRAM

Using Arduino IDE, we extract the smoke level values from the surrounding using the MQ2 smoke sensor via our hardware configuration. The value is then put in a If-Else block, which will determine if the smoke levels are within the safety limits or not. If, it exceeds the safety limits, then Arduino will undergo a series of events. These events include, starting of exhaust fans, sounding of the emergency buzzer, indication with the help of appropriate LEDs, automatic opening of the emergency doors, along with starting of the water sprinkler system. Along with these, the arduino will also transfer the smoke data to the ESP8266 WiFi module. It will then trigger a web request to the IFTTT website. On the IFTTT website, we have created an applet, which will in turn send a SMS to the safety officer or the owner as soon as it receives the web request. Now, we have also added a 16x2 LCD display which will continuously display the smoke level on the factory floor. All these events will continue to take place, as long as the smoke level doesn't return to the normal values.

6 COMPONENTS REQUIRED AND THEIR BRIEF DESCRIPTION

1. Arduino IDE
2. IFTTT Web platform
3. Arduino UNO
4. ESP8266
5. Smoke Sensor MQ-02
6. LEDS
7. Buzzer
8. Servo Motor SG90
9. Breadboards
10. 220ohm resistors
11. Exhaust Fan
12. DC motor
13. 16x2 LCD Screen

Arduino IDE

We have used Arduino IDE to copy our codes to the Arduino UNO board. The Arduino IDE 2.0 is an open-source project, currently in its beta-phase. It is a big step from it's sturdy predecessor, Arduino IDE (1.8.13), and comes with revamped User Interface, improved board library manger, autocomplete feature and much more.
Features:

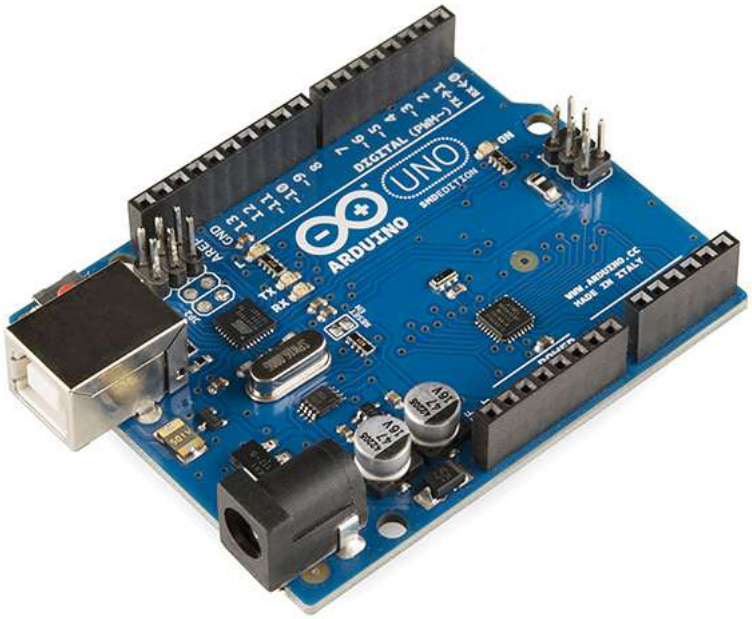
1. Arduino IDE provides us with a variety of sketch editing tools that allows us to use the entire functionality of the IDE quite easily and efficiently.
2. Arduino IDE also has a built in Serial Moniter that allows us to see the inputs obtained by various sensors.
3. The code is quite flexible and easy to understand and also can be shared quite easily.
4. It already includes a huge collection of libraries and also, newer libraries are being added to it quite frequently.
5. The IDE also includes a large collection of examples that helps in new users to get comfortable with the Arduino IDE.

IFTTT Web Platform

IFTTT is short for If This Then That, and is the best way to integrate apps, devices, and servicesIFTTT was founded on the belief that every thing works better together. Tech incompatibility has become challenging for anyone trying to build a smart home or create automatic routines in their life. IFTTT makes it easy.
Each service has a combination of triggers and actions that can be combined to create the automations that help you achieve your goals, be more efficient, and improve your smart home.

1. Choose your trigger
2. Choose your action
3. Name your Applet and hit Finish

Arduino UNO



We have used Arduino UNO microcontroller to support our project as it is quite efficient and affordable for our project. Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller

Features:

1. Microcontroller: ATmega328
2. Operating Voltage: 5V
3. Digital I/O Pins: 14 (of which 6 provide PWM output)
4. Analog Input Pins: 6
5. Flash Memory: 32 KB of which 0.5 KB used by bootloader
6. Clock Speed: 16 MHz

Smoke Sensor MQ-02

Gas Sensor(MQ2) module is useful for gas leakage detection (home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.
Features:
Wide detecting scope
Stable and long lifetime
Fast response and High sensitivity

Buzzer

The buzzer used with arduino programming is also known as the piezo buzzer. It is basically a tiny speaker that can sound a tony at a frequency we want. The buzzer produces sound based on reverse of the piezoelectric effect.Piezo ectricity is an effect where certain crystals will change shape when you apply electricity to them. By applying an electric signal at the right frequency, the crystal can make sound.

Micro Servo Motor SG90

Micro Servo Motor is a tiny, lightweight servo motor with high output power. It can rotate 180 degrees (90 degrees in each side) and can operate without any motor controller with feedback or gear box.
Features:

1. Weight: 9 g
2. Dimension: 22.2 x 11.8 x 31 mm approx.
3. Stall torque: 1.8 kgf·cm
4. Operating speed: 0.1 s/60 degree
5. Operating voltage: 4.8 V (5V)

7 PART 1 EXECUTION

Arduino Code

```
    #include<Servo.h>
//int pos;
//Servo Myservo;
void setup() {
  // put your setup code here, to run once:
  pinMode(A0, INPUT);//smoke sensor
  pinMode(7, OUTPUT);//led1
  pinMode(8, OUTPUT);//led2
  Serial.begin(9600);
  pinMode(3, OUTPUT);//exhaust fan
  pinMode(6, OUTPUT);//buzzer

  //Myservo.attach(10);//Servo motor

}

void loop() {
  // put your main code here, to run repeatedly:
  int a=analogRead(A0);
  Serial.println(a);
  delay(200);
  if(a>50){
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    digitalWrite(3,LOW);
    tone(6,1000);
  }
  else
  {
    digitalWrite(8,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(3,HIGH);
    noTone(6);

}
}
```

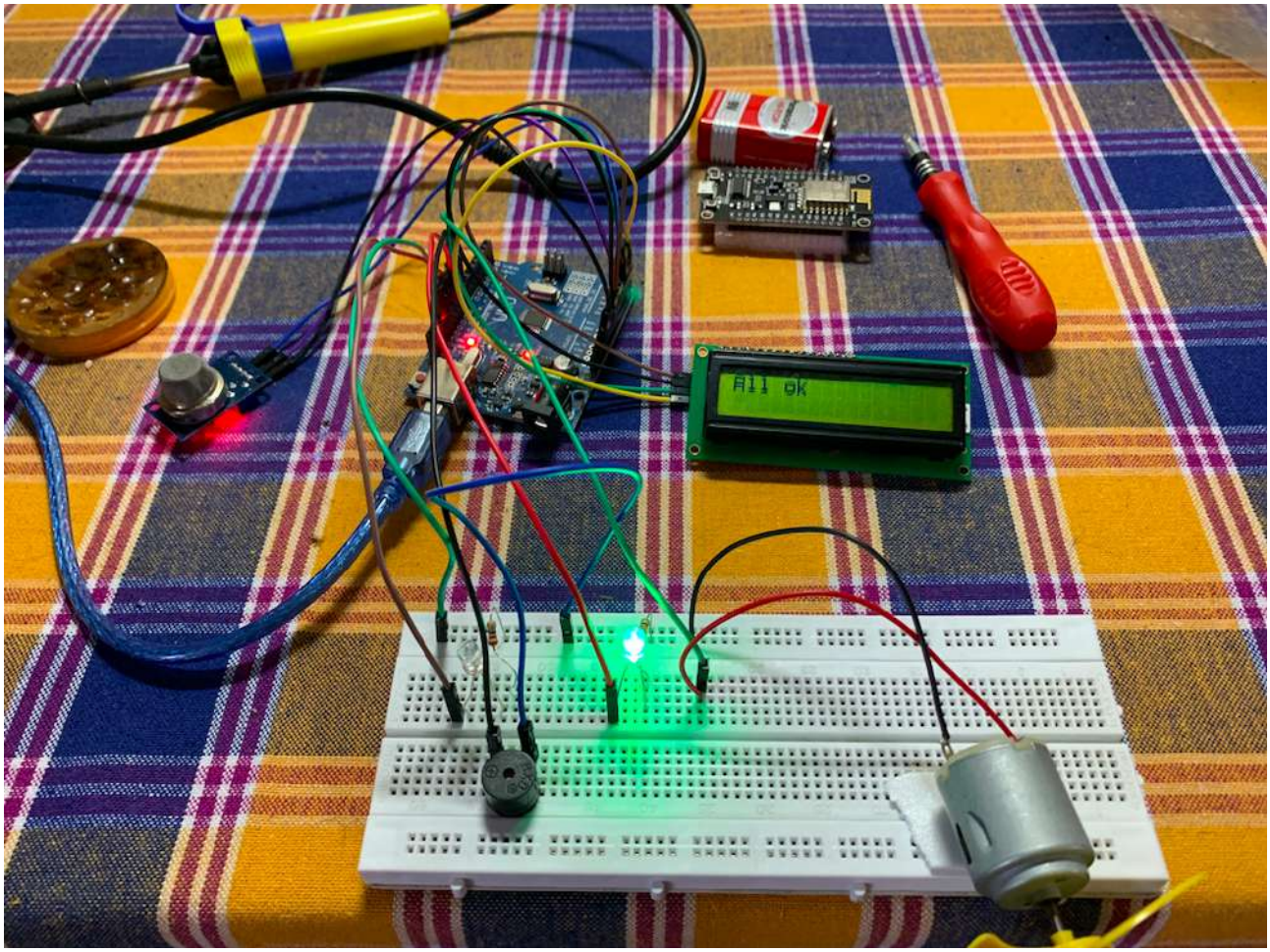


Figure 2: Arduino Model

NodeMCU Part

```
    #include <I2Cdev.h>

#include <LiquidCrystal_I2C.h>

//#include <LiquidCrystal_I2C.h>

#include<SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
//#include <LiquidCrystal_I2C.h>
#include<Wire.h>
LiquidCrystal_I2C lcd(0x3F,16,2);

const char* ssid = "Siddhanta";
const char* password = "121242121";

const char* host = "maker.ifttt.com";
const int httpsPort = 443;
const char* ch = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";

// Use web browser to view and copy
// SHA1 fingerprint of the certificate
const char* fingerprint = "616275faea5f64954af6090f59c90de71e6d66a3";
SoftwareSerial mySerial(D3,D4);
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  mySerial.begin(9600);
  Serial.println();
  Serial.print("connecting to ");
  Serial.println(ssid);
  // WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    //pinMode(smokeA0, INPUT);
    }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  lcd.begin();
  lcd.backlight();
```

```
}

void loop() {
  // put your main code here, to run repeatedly:
  String msg = mySerial.readStringUntil('\r');
  Serial.println(msg);
  int num;
  num=msg.toInt();

  if(num>50){
    WiFiClientSecure client;
        Serial.print("connecting to ");
        Serial.println(host);

        Serial.print("Using fingerprint: ");
        Serial.println(fingerprint);
        client.setFingerprint(fingerprint);

        if (!client.connect(host, httpsPort))
        {
          Serial.println("connection failed");
          return;
        }

        String url = "/trigger/Smoke_detected/json/with/key/bmKuHq_SOHNIQbf
        Serial.print("requesting URL: ");
        Serial.println(url);

        client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                     "Host: " + host + "\r\n" +
                     "User-Agent: BuildFailureDetectorESP8266\r\n" +
                     "Connection: close\r\n\r\n");

        Serial.println("request sent");
  }
  else{
    Serial.println("ALL ok");
    lcd.setCursor(0,0);
  lcd.print("hello");
    }

  }
```

OUTPUT VIDEO

https://drive.google.com/file/d/1fIi75fJ64ew-XUNrCZGFDQJPIVgnlv-M/view?usp=share_link

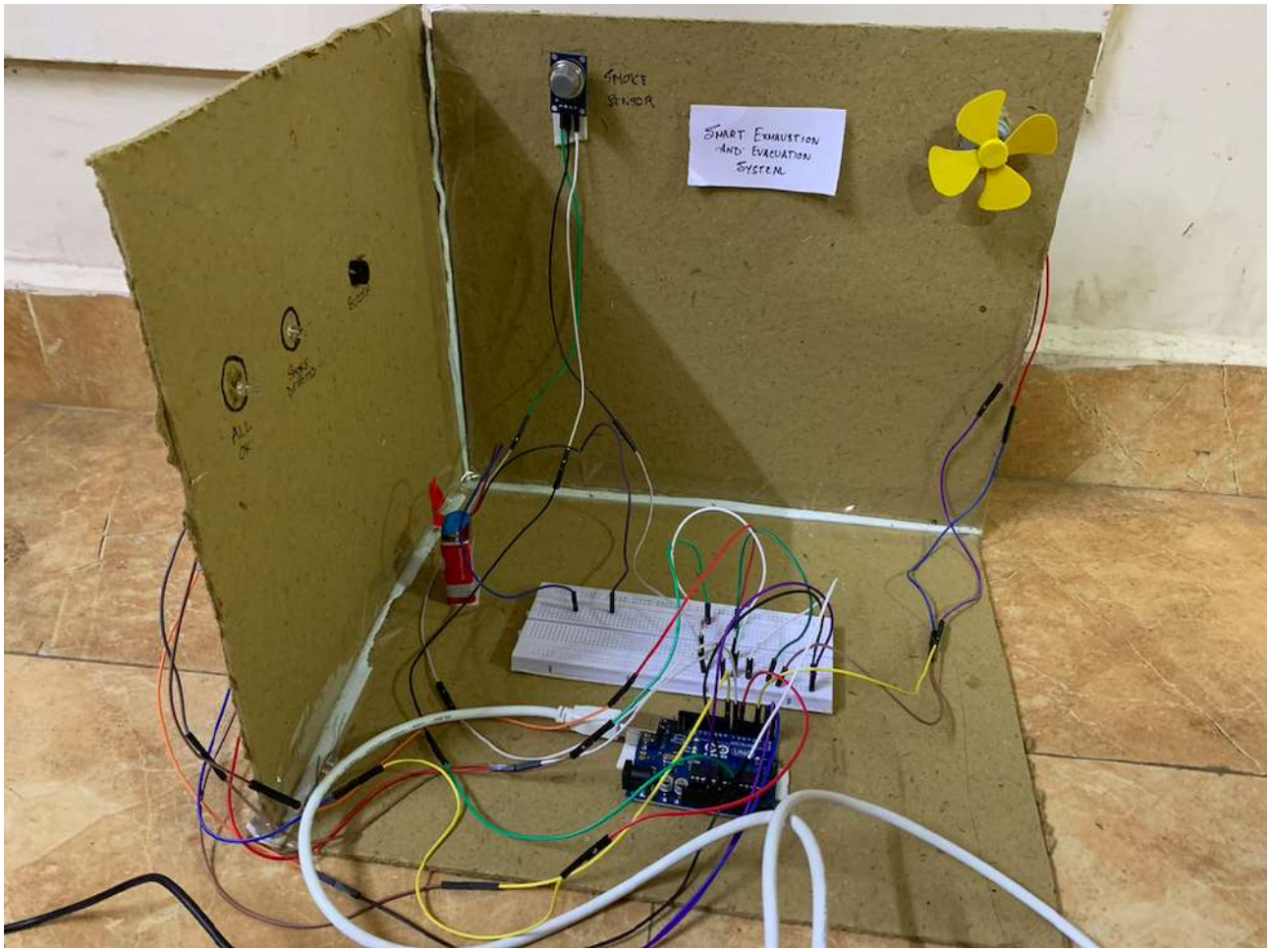


Figure 3: Final Model

8 FUTURE DEVELOPMENTS

Currently, we have developed a prototype using Arduino UNO and ESP8266. However, to make this industry oriented, we can make the same thing using a standard microcontroller, like 8051 family microcontrollers. Then we can have a greater power input, that will help in using industry grade components.

9 INFERENCE AND RESULTS

In this project, we have successfully demonstrated the working of a Smart Exhaustion and Evacuation system with the help of Arduino UNO and ESP8266. We have also created a real life model demonstrating the actual positions of all the componnents that are to be placed and also how they are to be connected with each other.
Finally, we tested the model in various situations and got affirmative results.

Final Working Link:
https://drive.google.com/file/d/1fIi75fJ64ew-XUNrCZGFDQJPIVgnlv-M/view?usp=share_link