

# Internet of Things Lab

## Digital Assignment 5

SREENIVASAN S

21BECO256

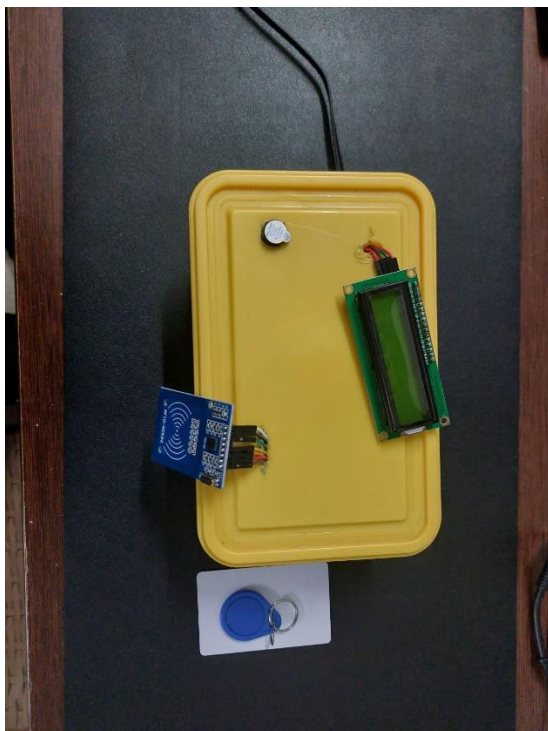
### Aim:

To construct a circuit to mark the attendance of students using an RFID tag and displaying it via a Google Sheets which will be displayed via a local website.

### Components Required:

Name	Quantity
ESP8266 Node MCU	1
Bread board	1
RFID Sensor	1
I2C LCD	1
RFID Tag	2
Active Buzzer	1
Jumper wire	multiple

### Circuit:



## Code:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const uint8_t fingerprint[20] = {0x56, 0xED, 0xC7, 0xDA, 0xBF, 0x51, 0x12,
0xC2, 0x79, 0x43, 0xC6, 0x01, 0xAB, 0xF7, 0x88, 0x98, 0x0F, 0x97, 0xB2, 0xB8};

LiquidCrystal_I2C lcd(0x27, 16, 2);

#define RST_PIN D3
#define SS_PIN D4
#define BUZZER D8

MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;
ESP8266WiFiMulti WiFiMulti;
MFRC522::StatusCode status;

int blockNum = 2;

byte bufferLen = 18;
byte readBlockData[18];

String data2;
const String data1 = "https://script.google.com/macros/s/AKfycbyo1Mz-m-u7PnQoNM0NjzkZNVeaixUZxwpqtGlgk3WG9BKaxVnYIdtA1ro5txANG1W-/exec?name=";

void setup()
{
    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print(" Attendance ");
    lcd.setCursor(0,1);
    lcd.print("Monitor");

    Serial.println();
    Serial.println();
}
```

```

Serial.println();

for (uint8_t t = 4; t > 0; t--)
{
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
}

WiFi.mode(WIFI_STA);

WiFiMulti.addAP("mahadevan", "mahadevan");

pinMode(BUZZER, OUTPUT);

SPI.begin();
}

void loop()
{
    mfrc522.PCD_Init();

    if ( ! mfrc522.PICC_IsNewCardPresent())
    {
        return;
    }

    if ( ! mfrc522.PICC_ReadCardSerial())
    {
        return;
    }

    Serial.println();
    Serial.println(F("Reading last data from RFID..."));
    ReadDataFromBlock(blockNum, readBlockData);

    Serial.println();
    Serial.print(F("Last data in RFID:"));
    Serial.print(blockNum);
    Serial.print(F(" --> "));
    for (int j=0 ; j<16 ; j++)
    {
        Serial.write(readBlockData[j]);
    }
}

```

```

}
Serial.println();
digitalWrite(BUZZER, HIGH);
delay(200);
digitalWrite(BUZZER, LOW);
delay(200);
digitalWrite(BUZZER, HIGH);
delay(200);
digitalWrite(BUZZER, LOW);

if ((WiFiMulti.run() == WL_CONNECTED))
{
    std::unique_ptr<BearSSL::WiFiClientSecure>client(new
BearSSL::WiFiClientSecure);

    client->setFingerprint(fingerprint);

    data2 = data1 + String((char*)readBlockData);
    data2.trim();
    Serial.println(data2);

    HTTPClient https;
    Serial.print(F("[HTTPS] begin...\n"));
    if (https.begin(*client, (String)data2))
    {
        // HTTP
        Serial.print(F("[HTTPS] GET...\n"));

        int httpCode = https.GET();

        if (httpCode > 0)
        {
            Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
            String data3=(char*)readBlockData;
            data3.trim();
            lcd.setCursor(0,0);
            lcd.print(data3);
            lcd.setCursor(0,1);
            lcd.print("Present");

        }
        else
        {
            Serial.printf("[HTTPS] GET... failed, error: %s\n",
https.errorToString(httpCode).c_str());
        }
    }
}

```

```

        https.end();
        delay(1000);
    }
    else
    {
        Serial.printf("[HTTPS} Unable to connect\n");
    }
}
}

void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
    for (byte i = 0; i < 6; i++)
    {
        key.keyByte[i] = 0xFF;
    }

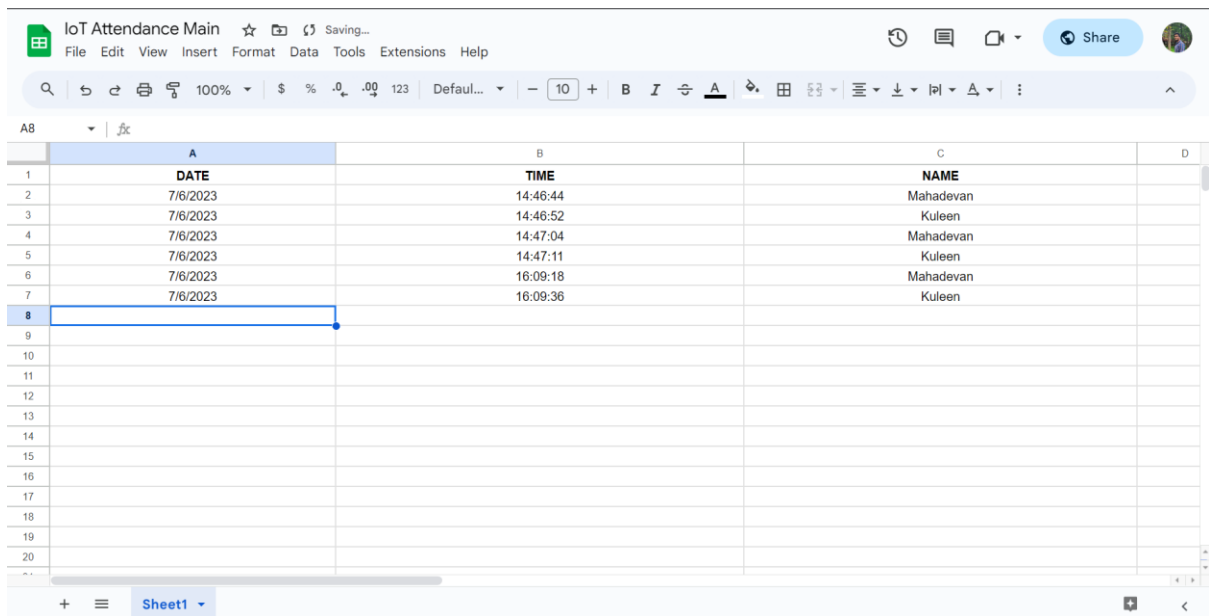
    status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNum,
    &key, &(mfrc522.uid));

    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("Authentication failed for Read: ");
        Serial.println(mfrc522.GetStatusCodeName(status));
        return;
    }
    else
    {
        Serial.println("Authentication success");
    }

    status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
    if (status != MFRC522::STATUS_OK)
    {
        Serial.print("Reading failed: ");
        Serial.println(mfrc522.GetStatusCodeName(status));
        return;
    }
    else
    {
        Serial.println("Block was read successfully");
    }
}
}

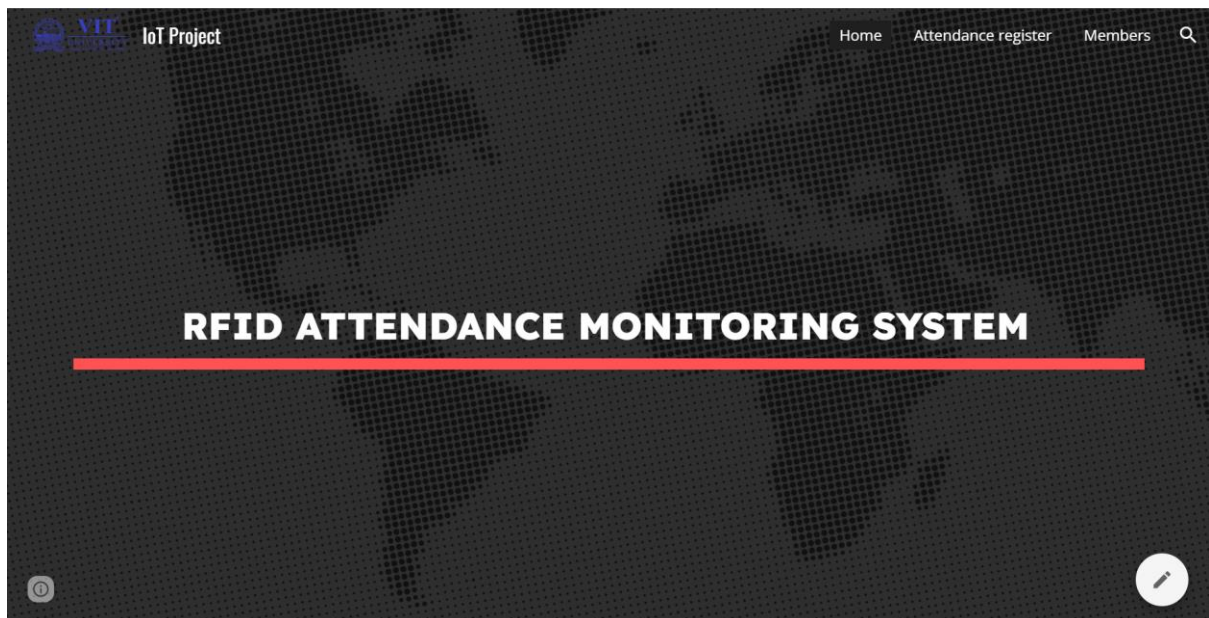
```

## Output:



The screenshot shows a Google Sheets spreadsheet titled "IoT Attendance Main". The spreadsheet has four columns: A (DATE), B (TIME), C (NAME), and D. The data is as follows:

	A	B	C	D
1	DATE	TIME	NAME	
2	7/6/2023	14:46:44	Mahadevan	
3	7/6/2023	14:46:52	Kuleen	
4	7/6/2023	14:47:04	Mahadevan	
5	7/6/2023	14:47:11	Kuleen	
6	7/6/2023	16:09:18	Mahadevan	
7	7/6/2023	16:09:36	Kuleen	
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				



IoT Attendance Main : Sheet1

DATE	TIME	NAME
7/6/2023	14:46:44	Mahadevan
7/6/2023	14:46:52	Kuleen
7/6/2023	14:47:04	Mahadevan
7/6/2023	14:47:11	Kuleen
7/6/2023	16:09:18	Mahadevan
7/6/2023	16:09:36	Kuleen

Published by [Google Sheets](#) – [Report Abuse](#) – Updated automatically every 5 minutes

## Result and Inference:

The code has been run and the circuit was simulated. The required output was observed. Whenever, the tag was brought near the sensor, it would beep the buzzer, and the name of the attendee, the date and the time of registering for attendance would be recorded into a google sheets. This value would be then sent to a website where it would be displayed and analysed.