ATTENDANCE MONITORING SYSTEM

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <SPI.h>
#include <MFRC522.h>
MFRC522 mfrc522(D4,D3);
#define WLAN_SSID      "ssid"
#define WLAN_PASS      "password"
#define AIO_SERVER     "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "username"
#define AIO_KEY        "key"
int ap;
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Publish photocell = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/c1");
void MQTT_connect();
void setup() {
  Serial.begin(115200);
  pinMode(D2,OUTPUT);
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init();
  delay(10);
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
 Serial.println();
 Serial.println("WiFi connected");
 Serial.println("IP address: "); Serial.println(WiFi.localIP());
}
uint32_t x=0;
void loop() {
 MQTT_connect();
 Serial.println("TAP YOUR CARD");
 // Look for new cards
 if ( ! mfrc522.PICC_IsNewCardPresent()) {
  return;
 }
 // Select one of the cards
 if ( ! mfrc522.PICC_ReadCardSerial()) {
  return;
 }

 //Reading from the card
 String tag = "";
 for (byte i = 0; i < mfrc522.uid.size; i++)
 {
  tag.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
  tag.concat(String(mfrc522.uid.uidByte[i], HEX));
 }
 tag.toUpperCase();
 Serial.println(tag.substring(1));
 if(tag.substring(1)=="33 DD AE 0F"&&ap==0)
 {
  digitalWrite(D2,HIGH);
  delay(500);
  digitalWrite(D2,LOW);
  Serial.println("DHARSHINI LOGIN");
  photocell.publish("DHARSHINI LOGIN");
  delay(3000);
  ap=1;
 }
 else if(tag.substring(1)=="33 DD AE 0F"&&ap==1)
```

```
 {
  digitalWrite(D2,HIGH);
  delay(500);
  digitalWrite(D2,LOW);
  Serial.println("DHARSHINI LOGOUT");
  photocell.publish("DHARSHINI LOGOUT");
  delay(3000);
  ap=0;
 }
}
void MQTT_connect() {
 int8_t ret;
 if (mqtt.connected()) {
  return;
 }
 Serial.print("Connecting to MQTT... ");
 uint8_t retries = 3;
 while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0) {
     while (1);
    }
 }
 Serial.println("MQTT Connected!");
}
```