#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

#include <RTClib.h>

#include <SPI.h>

#include <MFRC522.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Change the address to match your LCD

RTC\_DS3231 rtc;

Servo servo;

int servoPin = 6;

int buzzerPin = 8; // Connect the buzzer to pin 8

MFRC522 mfrc522(10, 9); // Define SS and RST pins

const int numAuthorized = 5; // 4 students and 1 teacher

const char\* authorizedNames[] = {"YAMINI", "SUKANIYA", "SUHAINA", "CHARU", "SIR AKBAR"};

byte authorizedUIDs[][4] = {

{0x07, 0xC8, 0xF4, 0x3D}, // Student1

{0x63, 0xC2, 0x8A, 0xDC}, // Student2

{0x8A, 0x6B, 0x7A, 0x07}, // Student3

{0xC3, 0xFF, 0xF4, 0xFA}, // Teacher

{0x8A, 0xE4, 0x7E, 0x07} // Student4 (New Student - Add the RFID tag here)

};

void setup() {

Serial.begin(9600);

servo.attach(servoPin);

servo.write(0);

lcd.init();

lcd.backlight();

Wire.begin();

rtc.begin();

lcd.setCursor(0, 0);

lcd.print("RFID BASED AUTH SYSTEM");

SPI.begin();

mfrc522.PCD\_Init();

pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as an output

digitalWrite(buzzerPin, LOW); // Initialize the buzzer to OFF state

}

void loop() {

DateTime now = rtc.now();

int currentTime = now.hour() \* 100 + now.minute();

// Display the real-time clock on the LCD

lcd.setCursor(0, 1);

lcd.print("Time: " + String(now.hour()) + ":" + String(now.minute()));

if ((currentTime >= 1000 && currentTime <= 2200)) { // Access from 10 AM to 10 PM

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {

if (checkUID(mfrc522.uid.uidByte)) {

unlockDoor();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Access Granted");

lcd.setCursor(0, 1);

lcd.print("Welcome, " + String(getAuthorizedName(mfrc522.uid.uidByte)));

activateBuzzer(); // Turn on the buzzer

delay(5000);

deactivateBuzzer(); // Turn off the buzzer

lockDoor();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("RFID BASED AUTH SYSTEM");

}

}

}

}

bool checkUID(byte\* uid) {

for (int i = 0; i < numAuthorized; i++) {

bool match = true;

for (int j = 0; j < 4; j++) {

if (uid[j] != authorizedUIDs[i][j]) {

match = false;

break;

}

}

if (match) {

return true;

}

}

return false;

}

String getAuthorizedName(byte\* uid) {

for (int i = 0; i < numAuthorized; i++) {

bool match = true;

for (int j = 0; j < 4; j++) {

if (uid[j] != authorizedUIDs[i][j]) {

match = false;

break;

}

}

if (match) {

return String(authorizedNames[i]);

}

}

return "Unknown";

}

void unlockDoor() {

servo.write(90);

}

void lockDoor() {

servo.write(0);

}

void activateBuzzer() {

digitalWrite(buzzerPin, HIGH); // Turn on the buzzer

}

void deactivateBuzzer() {

digitalWrite(buzzerPin, LOW); // Turn off the buzzer

}