#include <Adafruit\_Fingerprint.h>

#include <LiquidCrystal.h>

#include <EEPROM.h>

// Pin Definitions

#define MATCH\_BTN A4

#define ENROLL\_BTN A3

#define DEL\_OK\_BTN A2

#define CANDIDATE1 5

#define CANDIDATE2 4

#define CANDIDATE3 A1

#define RESULT\_BTN A0

#define BUZZER A5

#define YELLOW\_LED 7

#define GREEN\_LED 6

// Fingerprint Sensor

#define FINGERPRINT\_TX 2

#define FINGERPRINT\_RX 3

// Initialize LCD (RS, E, D4, D5, D6, D7)

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// Fingerprint Sensor

SoftwareSerial mySerial(FINGERPRINT\_TX, FINGERPRINT\_RX);

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

int voterCount = 0;

bool voted[25]; // Array to track voted IDs

int votes[3] = {0, 0, 0};

void setup() {

pinMode(MATCH\_BTN, INPUT\_PULLUP);

pinMode(ENROLL\_BTN, INPUT\_PULLUP);

pinMode(DEL\_OK\_BTN, INPUT\_PULLUP);

pinMode(CANDIDATE1, INPUT\_PULLUP);

pinMode(CANDIDATE2, INPUT\_PULLUP);

pinMode(CANDIDATE3, INPUT\_PULLUP);

pinMode(RESULT\_BTN, INPUT\_PULLUP);

pinMode(BUZZER, OUTPUT);

pinMode(YELLOW\_LED, OUTPUT);

pinMode(GREEN\_LED, OUTPUT);

lcd.begin(16, 2);

lcd.print("Voting Machine");

delay(2000);

lcd.clear();

finger.begin(57600);

if (finger.verifyPassword()) {

lcd.print("Sensor Connected");

} else {

lcd.print("Sensor Error");

while (1);

}

delay(2000);

lcd.clear();

resetData();

}

void loop() {

if (!digitalRead(ENROLL\_BTN)) {

enrollUser();

}

if (!digitalRead(MATCH\_BTN)) {

lcd.clear();

lcd.print("Place Finger");

delay(2000);

authenticateVoter();

}

if (!digitalRead(RESULT\_BTN)) {

displayResults();

}

}

void enrollUser() {

int id = voterCount + 1;

if (id > 25) {

lcd.clear();

lcd.print("Max Voters Reached");

delay(2000);

return;

}

lcd.clear();

lcd.print("Enroll ID: ");

lcd.print(id);

delay(2000);

lcd.clear();

lcd.print("Place Finger");

while (finger.getImage() != FINGERPRINT\_OK);

finger.image2Tz(1);

lcd.clear();

lcd.print("Remove Finger");

delay(2000);

lcd.clear();

lcd.print("Place Again");

while (finger.getImage() != FINGERPRINT\_OK);

finger.image2Tz(2);

if (finger.createModel() == FINGERPRINT\_OK) {

if (finger.storeModel(id) == FINGERPRINT\_OK) {

lcd.clear();

lcd.print("Enrolled!");

voterCount++;

EEPROM.update(0, voterCount);

} else {

lcd.clear();

lcd.print("Enroll Failed");

}

} else {

lcd.clear();

lcd.print("Mismatch Error");

}

delay(2000);

}

void authenticateVoter() {

int attempts = 3;

int userID = -1;

while (attempts--) {

if (finger.getImage() == FINGERPRINT\_OK) {

finger.image2Tz(1);

if (finger.fingerFastSearch() == FINGERPRINT\_OK) {

userID = finger.fingerID;

break;

}

}

}

if (userID == -1) {

lcd.clear();

lcd.print("No Match Found");

delay(2000);

return;

}

if (EEPROM.read(userID) == 1) {

lcd.clear();

lcd.print("Already Voted");

digitalWrite(BUZZER, HIGH);

delay(5000);

digitalWrite(BUZZER, LOW);

return;

}

lcd.clear();

lcd.print("Authorized Voter");

digitalWrite(GREEN\_LED, HIGH);

delay(2000);

digitalWrite(GREEN\_LED, LOW);

lcd.clear();

lcd.print("Vote: 1, 2, 3");

while (true) {

if (!digitalRead(CANDIDATE1)) {

votes[0]++;

EEPROM.update(1, votes[0]);

break;

}

if (!digitalRead(CANDIDATE2)) {

votes[1]++;

EEPROM.update(2, votes[1]);

break;

}

if (!digitalRead(CANDIDATE3)) {

votes[2]++;

EEPROM.update(3, votes[2]);

break;

}

}

EEPROM.update(userID, 1);

lcd.clear();

lcd.print("Vote Recorded");

delay(2000);

}

void displayResults() {

lcd.clear();

lcd.print("Results:");

delay(2000);

lcd.clear();

int c1 = votes[0];

int c2 = votes[1];

int c3 = votes[2];

lcd.print("C1: "); lcd.print(c1);

delay(2000);

lcd.clear();

lcd.print("C2: "); lcd.print(c2);

delay(2000);

lcd.clear();

lcd.print("C3: "); lcd.print(c3);

delay(2000);

lcd.clear();

int maxVotes = max(c1, max(c2, c3));

if (maxVotes == c1) lcd.print("Winner: C1");

else if (maxVotes == c2) lcd.print("Winner: C2");

else if (maxVotes == c3) lcd.print("Winner: C3");

delay(3000);

}

void resetData() {

voterCount = 0;

for (int i = 0; i < 25; i++) {

voted[i] = false;

EEPROM.update(i + 10, 0);

}

votes[0] = 0;

votes[1] = 0;

votes[2] = 0;

EEPROM.update(1, 0);

EEPROM.update(2, 0);

EEPROM.update(3, 0);

}