<https://www.instructables.com/Biometric-Door-Lock-Security-System-Using-Arduino-/>

Another code able to use

/\*  
\* Hello Friends Welcome To Techno-E-Solution  
\* Here Is Code For Bio-Metric Security System  
\*/#includeLiquidCrystal lcd(13,12,11,10,9,8);  
#include  
SoftwareSerial fingerPrint(2, 3);  
#include  
Servo myServo;  
#include  
uint8\_t id;  
Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&fingerPrint);  
#define enroll 14  
#define del 15  
#define up 16  
#define down 17  
#define openLight 6  
#define closeLight 7  
#define servoPin 5  
void setup()  
{  
delay(1000);  
myServo.attach(servoPin);  
myServo.write(180);  
pinMode(enroll, INPUT\_PULLUP);  
pinMode(up, INPUT\_PULLUP);  
pinMode(down, INPUT\_PULLUP);  
pinMode(del, INPUT\_PULLUP);  
pinMode(openLight, OUTPUT);  
pinMode(closeLight, OUTPUT);  
lcd.begin(16,2);  
lcd.print(” BIOMETRIC “);  
lcd.setCursor(0,1);  
lcd.print(“SECURITY SYSTEM”);  
delay(2000);  
lcd.clear();  
lcd.print(“TechnoEsolution”);  
delay(2000);  
finger.begin(57600);  
Serial.begin(9600);  
lcd.clear();  
lcd.print(“Finding Module”);  
lcd.setCursor(0,1);  
delay(1000);  
if (finger.verifyPassword())  
{  
Serial.println(“Found fingerprint sensor!”);  
lcd.clear();  
lcd.print(“Found Module “);  
delay(1000);  
}  
else  
{  
Serial.println(“Did not find fingerprint sensor :(“);  
lcd.clear();  
lcd.print(“Module not Found”);  
lcd.setCursor(0,1);  
lcd.print(“Check Connections”);  
while (1);  
}  
}  
void loop()  
{  
lcd.setCursor(0,0);  
lcd.print(“Press UP/Down “);  
lcd.setCursor(0,1);  
lcd.print(“to start System”);  
digitalWrite(closeLight, HIGH);  
if(digitalRead(up)==0 || digitalRead(down)==0)  
{  
for(int i=0;i<5;i++)  
{  
lcd.clear();  
lcd.print(“Place Finger”);  
delay(2000);  
int result=getFingerprintIDez();  
if(result>=0)  
{  
digitalWrite(openLight, HIGH);  
digitalWrite(closeLight, LOW);  
lcd.clear();  
lcd.print(“Allowed”);  
lcd.setCursor(0,1);  
lcd.print(“Gete Opened “);  
myServo.write(0);  
delay(5000);  
myServo.write(180);  
digitalWrite(closeLight, HIGH);  
digitalWrite(openLight, LOW);  
lcd.setCursor(0,1);  
lcd.print(“Gate Closed “);  
return;  
}  
}  
}  
checkKeys();  
delay(1000);  
}  
void checkKeys()  
{  
if(digitalRead(enroll) == 0)  
{  
lcd.clear();  
lcd.print(“Please Wait”);  
delay(1000);  
while(digitalRead(enroll) == 0);  
Enroll();  
}  
else if(digitalRead(del) == 0)  
{  
lcd.clear();  
lcd.print(“Please Wait”);  
delay(1000);  
delet();  
}  
}  
void Enroll()  
{  
int count=0;  
lcd.clear();  
lcd.print(“Enroll Finger “);  
lcd.setCursor(0,1);  
lcd.print(“Location:”);  
while(1)  
{  
lcd.setCursor(9,1);  
lcd.print(count);  
if(digitalRead(up) == 0)  
{  
count++;  
if(count>25)  
count=0;  
delay(500);  
}  
else if(digitalRead(down) == 0)  
{  
count–;  
if(count<0)  
count=25;  
delay(500);  
}  
else if(digitalRead(del) == 0)  
{  
id=count;  
getFingerprintEnroll();  
return;  
}  
else if(digitalRead(enroll) == 0)  
{  
return;  
}  
}  
}  
void delet()  
{  
int count=0;  
lcd.clear();  
lcd.print(“Delete Finger “);  
lcd.setCursor(0,1);  
lcd.print(“Location:”);  
while(1)  
{  
lcd.setCursor(9,1);  
lcd.print(count);  
if(digitalRead(up) == 0)  
{  
count++;  
if(count>25)  
count=0;  
delay(500);  
}  
else if(digitalRead(down) == 0)  
{  
count–;  
if(count<0)  
count=25;  
delay(500);  
}  
else if(digitalRead(del) == 0)  
{  
id=count;  
deleteFingerprint(id);  
return;  
}  
else if(digitalRead(enroll) == 0)  
{  
return;  
}  
}  
}  
uint8\_t getFingerprintEnroll()  
{  
int p = -1;  
lcd.clear();  
lcd.print(“finger ID:”);  
lcd.print(id);  
lcd.setCursor(0,1);  
lcd.print(“Place Finger”);  
delay(2000);  
while (p != FINGERPRINT\_OK)  
{  
p = finger.getImage();  
switch (p)  
{  
case FINGERPRINT\_OK:  
Serial.println(“Image taken”);  
lcd.clear();  
lcd.print(“Image taken”);  
break;  
case FINGERPRINT\_NOFINGER:  
Serial.println(“No Finger”);  
lcd.clear();  
lcd.print(“No Finger”);  
break;  
case FINGERPRINT\_PACKETRECIEVEERR:  
Serial.println(“Communication error”);  
lcd.clear();  
lcd.print(“Comm Error”);  
break;  
case FINGERPRINT\_IMAGEFAIL:  
Serial.println(“Imaging error”);  
lcd.clear();  
lcd.print(“Imaging Error”);  
break;  
default:  
Serial.println(“Unknown error”);  
lcd.clear();  
lcd.print(“Unknown Error”);  
break;  
}  
}  
// OK success!  
p = finger.image2Tz(1);  
switch (p) {  
case FINGERPRINT\_OK:  
Serial.println(“Image converted”);  
lcd.clear();  
lcd.print(“Image converted”);  
break;  
case FINGERPRINT\_IMAGEMESS:  
Serial.println(“Image too messy”);  
lcd.clear();  
lcd.print(“Image too messy”);  
return p;  
case FINGERPRINT\_PACKETRECIEVEERR:  
Serial.println(“Communication error”);  
lcd.clear();  
lcd.print(“Comm Error”);  
return p;  
case FINGERPRINT\_FEATUREFAIL:  
Serial.println(“Could not find fingerprint features”);  
lcd.clear();  
lcd.print(“Feature Not Found”);  
return p;  
case FINGERPRINT\_INVALIDIMAGE:  
Serial.println(“Could not find fingerprint features”);  
lcd.clear();  
lcd.print(“Feature Not Found”);  
return p;  
default:  
Serial.println(“Unknown error”);  
lcd.clear();  
lcd.print(“Unknown Error”);  
return p;  
}