```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial mySerial(2, 3);
const int buttonPin1 = 4;
const int buttonPin2 = 5;
const int buttonPin3 = 6;
const int buzzer = 7;
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int id =0,previous_voter_id = 0, vote_taken = 0;
int party_1_count=0,party_2_count=0,party_3_count=0;
String winner_name = "";
void setup()
{
  pinMode(buzzer, OUTPUT);
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
    // initialize the lcd
  lcd.init();
    // Turn on the Backlight
  lcd.backlight();
  Serial.begin(9600);
```

```
 while (!Serial);  // For Yun/Leo/Micro/Zero/...
 delay(100);
 Serial.println("\n\nAdafruit finger detect test");

 // set the data rate for the sensor serial port
 finger.begin(57600);

 if (finger.verifyPassword()) {
   Serial.println("Found fingerprint sensor!");
 } else {
   Serial.println("Did not find fingerprint sensor :(");
   while (1) { delay(1); }
 }

 finger.getTemplateCount();
 Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
 Serial.println("Waiting for valid finger...");
   lcd.clear();
 // Set cursor (Column, Row)
 lcd.setCursor(0, 0);
 lcd.print("Smart Electronic");
 lcd.setCursor(0,1);
 lcd.print("Voting Machine");
 delay(3000);
}

void loop()                  // run over and over again
```

```
{

  // Clear the display buffer
  vote_taken = 0;
  lcd.clear();
  // Set cursor (Column, Row)
  lcd.setCursor(0, 0);
  lcd.print("Please place your");
  lcd.setCursor(0,1);
  lcd.print("finger");
  delay(100);
  id = getFingerprintIDez();
  if(id > 0)
  {
    // Clear the display buffer
  lcd.clear();
  // Set cursor (Column, Row)
  lcd.setCursor(0, 0);
  lcd.print("Your Voter ID");
  lcd.setCursor(0,1);
  lcd.print(id);
  delay(2000);
  if(id == 4)
  {
      if((party_1_count > party_2_count) && ((party_1_count > party_3_count)))
      {
        winner_name = "BJP";
      }
  else if((party_2_count > party_1_count) && ((party_2_count > party_3_count)))
  {
   winner_name = "NCP";
  }
  else
  {
   winner_name = "Congress";
  }
 // Clear the display buffer
 lcd.clear();
 // Set cursor (Column, Row)
 lcd.setCursor(0, 0);
 lcd.print("winner party");
 lcd.setCursor(0,1);
 lcd.print(winner_name);
 while(1);
}
if(previous_voter_id != id)
{
 do
 {
 // Clear the display buffer
 lcd.clear();
 // Set cursor (Column, Row)
 lcd.setCursor(0, 0);
 lcd.print("Give Your vote");
 lcd.setCursor(0,1);
 lcd.print("Press Button");
```

```
delay(500);
previous_voter_id = id;
buttonState1 = digitalRead(buttonPin1);
delay(10);
buttonState2 = digitalRead(buttonPin2);
delay(10);
buttonState3 = digitalRead(buttonPin3);
delay(10);
if (buttonState1 == HIGH)
{
 party_1_count = party_1_count +1;
 vote_taken = 1;
}
else if(buttonState2 == HIGH)
{
 party_2_count = party_2_count +1;
  vote_taken = 1;
}
else if(buttonState3 == HIGH)
{
 party_3_count = party_3_count +1;
 vote_taken = 1;
}
else
{
 vote_taken = 0;
}
if(vote_taken == 1)
```

```
 {
   // Clear the display buffer
   lcd.clear();
   // Set cursor (Column, Row)
   lcd.setCursor(0, 0);
   lcd.print("Thanks for your");
   lcd.setCursor(0,1);
   lcd.print("vote");
   delay(200);
   digitalWrite(buzzer, HIGH);   // turn the LED on (HIGH is the voltage level)
   delay(1000);                  // wait for a second
   digitalWrite(buzzer, LOW);    // turn the LED off by making the voltage LOW
   delay(1000);
 }
 }while(vote_taken == 0);
}
else
{
    // Clear the display buffer
lcd.clear();
// Set cursor (Column, Row)
lcd.setCursor(0, 0);
lcd.print("Duplicate Vote");
lcd.setCursor(0,1);
lcd.print("buzzer on");
delay(2000);
digitalWrite(buzzer, HIGH);   // turn the LED on (HIGH is the voltage level)
delay(1000);                  // wait for a second
```

```
  digitalWrite(buzzer, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);
  digitalWrite(buzzer, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(buzzer, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);
  digitalWrite(buzzer, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(buzzer, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);
  }
  }
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
  }

  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);

  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
```