```
#include<EEPROM.h>
#include<LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#include <Adafruit_Fingerprint.h>
uint8_t id;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial);

#define enroll 14
#define del 15
#define up 16
#define down 17
#define match 18
#define indVote 6

#define sw1 5
#define sw2 4
#define sw3 3
#define resultsw 2
#define indFinger 7
#define buzzer 19
#define records 25
int vote1,vote2,vote3;

int flag;

void setup()
{
    delay(1000);
    pinMode(enroll, INPUT_PULLUP);
    pinMode(up, INPUT_PULLUP);
    pinMode(down, INPUT_PULLUP);
    pinMode(del, INPUT_PULLUP);
    pinMode(match, INPUT_PULLUP);
    pinMode(sw1, INPUT_PULLUP);
    pinMode(sw2, INPUT_PULLUP);
    pinMode(sw3, INPUT_PULLUP);
    pinMode(resultsw, INPUT_PULLUP);
    pinMode(buzzer, OUTPUT);
    pinMode(indVote, OUTPUT);
    pinMode(indFinger, OUTPUT);

lcd.begin(16,2);
  if(digitalRead(resultsw) ==0)
  {
      for(int i=0;i<records;i++)
      EEPROM.write(i+10,0xff);
    EEPROM.write(0,0);
    EEPROM.write(1,0);
    EEPROM.write(2,0);
    lcd.clear();
    lcd.print("System Reset");
    delay(1000);
}

  lcd.clear();
  lcd.print("Voting Machine");
  lcd.setCursor(0,1);
  lcd.print("by Finger Print");
  delay(2000);
  lcd.clear();
  lcd.print("Team members:");
  lcd.setCursor(0,1);
  lcd.print("Ashwak Khan");
  lcd.print("Aruneshwaran");
  lcd.setCursor(0,1);
  lcd.print("Divakaran");
  delay(2000);

if(EEPROM.read(0) == 0xff)
 EEPROM.write(0,0);

   if(EEPROM.read(1) == 0xff)
 EEPROM.write(1,0);

   if(EEPROM.read(1) == 0xff)
 EEPROM.write(1,0);



  //finger.begin(57600);
  Serial.begin(57600);
  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);
   }

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Cn1");
  lcd.setCursor(4,0);
  lcd.print("Cn2");
  lcd.setCursor(8,0);
  lcd.print("Cn3");
  lcd.setCursor(12,0);
  lcd.print("Cn4");

  lcd.setCursor(0,1);
  vote1=EEPROM.read(0);
  lcd.print(vote1);
  lcd.setCursor(6,1);
  vote2=EEPROM.read(1);
  lcd.print(vote2);
  lcd.setCursor(12,1);
  vote3=EEPROM.read(2);
  lcd.print(vote3);
  delay(2000);
}

void loop()
{
 lcd.setCursor(0,0);
 lcd.print("Press Match Key ");
 lcd.setCursor(0,1);
 lcd.print("to verify user");

 digitalWrite(indVote, LOW);
 digitalWrite(indFinger, LOW);
 if(digitalRead(match)==0)
  {
  digitalWrite(buzzer, HIGH);
  delay(200);
  digitalWrite(buzzer, LOW);
  digitalWrite(indFinger, HIGH);
  for(int i=0;i<3;i++)
  {
     lcd.clear();
    lcd.print("Place Finger");
    delay(2000);
    int result=getFingerprintIDez();
    if(result>=0)
    {
      flag=0;
        for(int i=0;i<records;i++)
        {
          if(result == EEPROM.read(i+10))
          {
             lcd.clear();
              lcd.print("Authorised Voter");
              lcd.setCursor(0,1);
              lcd.print("Please Wait....");
              delay(1000);
              Vote();
              EEPROM.write(i+10, 0xff);
              flag=1;
              return;
          }
        }

        if(flag == 0)
        {
        lcd.clear();
        lcd.print("Already Voted");
        //lcd.setCursor(0,1);
        //lcd.print("")
        digitalWrite(buzzer, HIGH);
        delay(5000);
        digitalWrite(buzzer, LOW);
        return;
        }
     }
   }
   lcd.clear();
 }
 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("Enter Finger ID:");

  while(1)
  {
   lcd.setCursor(0,1);
    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();
          for(int i=0;i<records;i++)
          {
            if(EEPROM.read(i+10) == 0xff)
            {
              EEPROM.write(i+10, id);
              break;
            }
          }
          return;
     }

       else if(digitalRead(enroll) == 0)
     {
          return;
     }
 }
}

void delet()
{
   int count=0;
   lcd.clear();
   lcd.print("Enter Finger ID");

   while(1)
   {
    lcd.setCursor(0,1);
     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);
          for(int i=0;i<records;i++)
          {
            if(EEPROM.read(i+10) == id)
            {
              EEPROM.write(i+10, 0xff);
              break;
            }
          }
          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;
}

//Serial.println("Remove finger");
lcd.clear();
lcd.print("Remove Finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
//Serial.print("ID "); //Serial.println(id);
p = -1;
//Serial.println("Place same finger again");
 lcd.clear();
    lcd.print("Place Finger");
    lcd.setCursor(0,1);
    lcd.print("   Again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    //Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    //Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    //Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    //Serial.println("Imaging error");
    break;
  default:
    //Serial.println("Unknown error");
    return;
  }
}

// OK success!

p = finger.image2Tz(2);
```

```
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!
//Serial.print("Creating model for #");  //Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
  //Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  //Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  //Serial.println("Fingerprints did not match");
  return p;
} else {
  //Serial.println("Unknown error");
  return p;
}

//Serial.print("ID "); //Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  //Serial.println("Stored!");
  lcd.clear();
  lcd.print("Stored!");
  delay(2000);
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    //Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    //Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    //Serial.println("Error writing to flash");
    return p;
  }
  else {
    //Serial.println("Unknown error");
    return p;
  }
}

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)
  {
   lcd.clear();
   lcd.print("Finger Not Found");
   lcd.setCursor(0,1);
   lcd.print("Try Later");
   delay(2000);
  return -1;
  }
  // found a match!
  //Serial.print("Found ID #");
  //Serial.print(finger.fingerID);
  return finger.fingerID;
}

uint8_t deleteFingerprint(uint8_t id)
{
  uint8_t p = -1;
  lcd.clear();
  lcd.print("Please wait");
  p = finger.deleteModel(id);
  if (p == FINGERPRINT_OK)
  {
    //Serial.println("Deleted!");
    lcd.clear();
    lcd.print("Figer Deleted");
    lcd.setCursor(0,1);
    lcd.print("Successfully");
    delay(1000);
  }

  else
  {
    //Serial.print("Something Wrong");
    lcd.clear();
    lcd.print("Something Wrong");
    lcd.setCursor(0,1);
    lcd.print("Try Again Later");
    delay(2000);
    return p;
  }
}

void Vote()
{
  lcd.clear();
  lcd.print("Please Place");
  lcd.setCursor(0,1);
  lcd.print("Your Vote");
  digitalWrite(indVote, HIGH);
  digitalWrite(indFinger, LOW);
  digitalWrite(buzzer, HIGH);
  delay(500);
  digitalWrite(buzzer, LOW);
  delay(1000);
  while(1)
  {
        if(digitalRead(sw1)==0)
        {
          vote1++;
          voteSubmit(1);
          EEPROM.write(0, vote1);
          while(digitalRead(sw1)==0);
          return;
        }
```

```
if(digitalRead(sw2)==0)
{
  vote2++;
    voteSubmit(2);
    EEPROM.write(1, vote2);
  while(digitalRead(sw2)==0);
  return;
}
if(digitalRead(sw3)==0)
{
  vote3++;
     voteSubmit(3);
     EEPROM.write(2, vote3);
  while(digitalRead(sw3)==0);
  return;
}

 if(digitalRead(resultsw)==0)
 {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Can1");
      lcd.setCursor(6,0);
      lcd.print("Can2");
      lcd.setCursor(12,0);
      lcd.print("Can3");
      for(int i=0;i<3;i++)
      {
        lcd.setCursor(i*6,1);
        lcd.print(EEPROM.read(i));
      }
      delay(2000);
   int vote=vote1+vote2+vote3;
   if(vote)
   {
    if((vote1 > vote2 && vote1 > vote3))
    {
     lcd.clear();
     lcd.print("Can1 Wins");
     delay(2000);
     lcd.clear();
    }
    else if(vote2 > vote1 && vote2 > vote3)
    {
     lcd.clear();
     lcd.print("Can2 Wins");
```

```
           delay(2000);
           lcd.clear();
          }
         else if((vote3 > vote1 && vote3 > vote2))
          {
           lcd.clear();
           lcd.print("Can3 Wins");
           delay(2000);
           lcd.clear();
          }

        else
        {
           lcd.clear();
           lcd.print("   Tie Up Or   ");
           lcd.setCursor(0,1);
           lcd.print("   No Result   ");
           delay(1000);
           lcd.clear();
        }

       }
        else
        {
           lcd.clear();
           lcd.print("No Voting....");
           delay(1000);
           lcd.clear();
        }
        vote1=0;vote2=0;vote3=0;vote=0;
        lcd.clear();
        return;
      }
  }
   digitalWrite(indVote, LOW);
}

void voteSubmit(int cn)
{
  lcd.clear();
  if(cn == 1)
      lcd.print("Can1");
  else if(cn == 2)
      lcd.print("Can2");
  else if(cn == 3)
      lcd.print("Can3");
```

```
 lcd.setCursor(0,1);
 lcd.print("Vote Submitted");
 digitalWrite(buzzer , HIGH);
 delay(1000);
 digitalWrite(buzzer, LOW);
 digitalWrite(indVote, LOW);
 return;
}
```