USHA KUMARI GRIET

fingerprint voting system report final.docx

Gokaraju Rangaraju Institute of Engineering and Technology

Document Details

Submission ID
trn:oid:::3618:96643195

Submission Date
May 19, 2025, 12:52 PM GMT+5:30

Download Date
May 19, 2025, 12:55 PM GMT+5:30

File Name
fingerprint voting system report final.docx

File Size
2.5 MB

46 Pages

5,068 Words

29,991 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

A real time project report on

FINGERPRINT VOTING SYSTEM

Project report submitted in partial fulfillment of the requirement for the award of the Degree of B. Tech

By

GURRAM LOHITH KUMAR	**23245A0403**
KOTTURU SAI MANISH	**23245A0405**
SOHAN REDDY	**22241A0453**
POLLURI DATTA SAI	**22241A0444**

Under the Guidance

Of

DR.CH USHA KUMARI

Head Of The Department of Electronics and Communication Engineering

Gokaraju Rangaraju Institute of Engineering and Technology

Department of Electronics and Communication Engineering

(AICTE Approved; Autonomous under JNTU) Bachupally ,

Kukatpally, Hyderabad-500090

Gokaraju Rangaraju Institute of Engineering and Technology

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the thesis entitled "FINGERPRINT VOTING SYSTEM," submitted by Kotturu Sai Manish (23245A0405), Sohan Reddy (22241A0453), Polluri Datta Sai (22241A0444) and Gurram Lohith Kumar (23245A0403), in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering at Jawaharlal Nehru Technological University Hyderabad (JNTUH) during the academic year 2024–2025, is a bona fide record of work carried out under our guidance and supervision. The contents of this thesis, in full or in parts, have not been submitted to any other university or institution for the award of any degree or diploma.

DECLARATION

I hereby declare that the real time project entitled **"FINGERPRINT VOTING SYSTEM"** is the work done during the period from **Apr 2025 to May 2025** and is submitted in the partial fulfillment of the requirements for the award of Bachelor of Technology in Electronics and Communication Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other university or Institution for the award of any Degree or Diploma.

GURRAM LOHTH KUMAR 23245A0403
KOTTURU SAI MANISH 23245A0405
SOHAN REDDY 22241A0453
POLLURI DATTA SAI 22241A0444

ACKNOWLEDGEMENT

I gratefully thank my project guide **Dr Usha Kumari,** Department of Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology for her excellent guidance and timely suggestions throughout the work.

I wish to express my sincere gratitude to **Project Review Committee Members (Dr K Meenakshi, Dr Hima Bindu Valiveti)**, Department of Electronics and Communication Engineering for providing valuable guidance and suggestions during the course of work.

I would like to convey my sincere thanks to Dr. Ch Usha Kumari, Professor and Head of the Department, Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology for valuable suggestions and motivation during the work.

I convey my deep sense of gratitude to Dr. J. Praveen, Principal for the kind encouragement and support.I would like to thank all teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their kind support and suggestions which in turn helped me to complete the project successfully.

Finally, I would like to convey my sincere thanks to each and every person who have supported and helped me for the success of the project.

Gurram Lohith Kumar(23245A0405)

Kotturu Sai Manish (23245A0405)

Sohan Reddy (22241A0453)

Polluri Datta Sai (22241A0444)

TABLE OF CONTENTS

ABSTRACT

Voter authorization and authentication are essential components for ensuring the security of the voting process. Traditional methods for verifying voter identity and stopping fraudulent voting are often inefficient and subject to human error. In order to overcome these challenges, we suggest a fingerprint-based voting system that combines microcontroller technology with biometric authentication, especially using Aadhar UIDAI data for validation. Before letting the voters to vote this system checks weather the voter is eligible or not, here the system checks in the database weather the scanned fingerprint exists in the database if it does exists then the system will not allow the person to vote in the other case if the fingerprint is not existing then it allows the voter to vote and add the fingerprint to database .Hence this system ensures that no malpractice is done.

The fingerprint-based voting system combines Internet of Things (IoT) technology, which improves security, and efficiency, by utilizing the biometric data from the Aadhar UIDAI. Real-time data transfer is possible by the Internet of Things component, which links the voting devices to a central server. By doing away with the need for traditional identification documents and decreasing the possibility of voter fraud, this method provides a safe, easy-to use voting process. Only valid, registered voters, as confirmed by Aadhar's UIDAI database, are permitted to cast ballots thanks to the system.

LIST OF FIGURES

LIST OF ACRONYMS

- **MCU: Micro Controller Unit**
- **LCD : Liquid Crystal Display**
- **TX : Transmitter**
- **RX : Receiver**

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Making voting safe and equal has been a challenge recently, especially due to the rise in cases of double and false voting. In order to tackle this, we created a voting system that checks voters using their fingerprints and Aadhaar. The concept is simple: only people who are registered and have a legitimate fingerprint can cast a ballot. To confirm our identities, we used a fingerprint sensor and a Arduino UNO controller. The user can use push buttons to vote for different candidates after confirmation. During the process, a buzzer provides prompt feedback. Voting will be safer, clearer, and more effective with this system, especially in rural regions where manual systems may not work.

The system is a suitable solution to voting problems because it uses biometric recognition and simple electronics. Because Arduino UNO UNO is small, reasonably priced, and works well with other parts like the LCD and fingerprint sensor, we used it. The components are connected in a way that makes the system neat and easy to read, and we program using the Arduino IDE. The fingerprint stops fraudulent or duplicate voting because it is connected to an Aadhaar ID. Despite being merely a prototype, it shows how the system can secure digital voting without being unduly complex or expensive. We learned a lot from this project, which also shows how technology can improve everyday events like elections.

1.2 FEATURES

- Biometric Authentication
- Aadhar Integration
- Secure Voting
- Real-time feedback
- Lcd Display
- Push Button Input
- ARDUINO UNO Controller
- Simple Design
- User friendly
- Low cost
- Offline Operation

1.3 PROBLEM DEFINITION

The absence of an identity verification system is one of the main issues with the present polling place. For the manual verification of votes, this can result in two and lengthy lines. The labor-intensive of the past result in plenty of paperwork and error counting. Voting is uncertain and impractical in some places, especially in rural areas. Obviously, we need a safer, better, and more secure voting system that is easy to use, quick, accurate, and accessible by all. Our project shows that we can do this by using the fingerprint identity linked to Aadhaar, ensuring that only legitimate voters are able to cast a single, valid ballot.

1.4 OBJECTIVE

The primary objective of this project is to use cheap, low-complexity hardware to create a secure electronic voting system with fingerprint and Aadhaar authentication. Voting will be made easier by the system's efficiency, security, and ease of use. The main goals are:

1. Biometric Verification:

The system features a fingerprint sensor for recognition to ensure that only the registered voter casts a ballot. This reduces the risk of someone voting twice or being as someone else

2. Linking Aadhaar:

The process is more valid since the voter's identity is confirmed using their Aadhaar information.

3. Immediate Feedback Process:

To keep the user informed of the process, an LCD display gives real-time messages and instructions, such as voting confirmation and status tests

4. **Push buttons are used by users to enter data:**

Voters used special push buttons to choose the candidates they wanted to support. Everyone, young or old, can do it easily and effortlessly.

5. Visual and auditory cues:

An alarm device that provides immediate audio feedback on successful or unsuccessful operations (such as voting or authentication success or failure) is a buzzer.

6. Design of Microcontrollers:

It runs by a Arduino, which allows efficient data processing and communication.

7. Simple Interface:

It requires little knowledge of technology and is easy to use for users of all skill levels. Economical and Adaptable: The system's use of open-source hardware and software makes it both affordable and scalable, making it suitable for use in community votes or important local elections.

CHAPTER 2
LITERATURE REVIEW

We studied how voting technologies have changed and what challenges still exist before creating our fingerprint and Aadhaar-enabled voting system. The key developments and weaknesses in earlier systems that affected the course of our project are discussed in this chapter.

2.1 Conventional Methods of Voting

Electronic voting machines (EVMs) and paper ballots have been used in elections for decades. even being well-known and frequently employed, these methods still have some major issues:

- Errors may occur when counting votes by hand.
- Weak identity checks allow for voter pretending to be someone or repeat voting.
- Verification of voter ID is typically done by hand, which isn't always accurate

These drawbacks highlight the need for automated, more secure systems.

2.2 Change to Online and Electronic Voting

There have been many attempts to move voting online or make it completely electronic as a result of the development of digital technology. Making voting easier and faster was the goal. The problem of voter authentication was not entirely resolved by many of these systems, though. Fraud and illegal access have a possibility in the absence of strong verification.

2.3 Voting Systems with Biometrics

Biometric systems, especially fingerprint recognition, have gained popularity as a solution to the identity problem. Banks and offices have already successfully used biometrics for safe access.

2.4 Verification Based on Aadhaar

Every Indian citizen has a unique identity due to Aadhaar, which is based on both biometric and social information. We can improve the accuracy of voter identity verification by connecting a fingerprint sensor to Aadhaar data. This avoids unauthorized users from accessing the system and helps ensure that each voter casts just one vote.

2.5 Issues in Present Systems

Even such advances, previous efforts frequently ran into issues like:

- No voter identity verification in real time.
- Insufficient user guidance during the voting process.
- Difficult user interfaces that were difficult for non-technical people to use. Costly or complex systems that are hard for common usage.

These issues show that more research needs to be done to create a smarter system.

2.6 Our Perspective on the Issue

The purpose of our project was to close those gaps. Our goal is to create a system that is:

- Safe with fingerprint verification.
- Connected to Aadhaar in order to verify the identity of every voter. Push buttons make selection simple.
- Responsive, featuring an instantaneous sound feedback buzzer.
- Economical and effective thanks to the Arduino UNO and the Arduino IDE for programming.

By integrating these components, our system is made to be secure, dependable, and easy to use a feature that could potentially be modified for future local or national use.

CHAPTER 3
PROPOSED METHODOLOGY

3.1 System overview

Basic Idea of the Project:

Nowadays, a lot of voting still happens using paper or basic EVMs, which can be misused or manipulated. So we thought, why not make something better using the tech we already have? By using fingerprint sensors and microcontrollers, we can make sure only real and registered people are allowed to vote. We also added a display and buttons to make it interactive, and a buzzer for feedback. All of this is controlled using Arduino UNO.

Figure 3.1: Fingerprint Voting System

3.2Requirement engineering

3.2.1 Hardware requirements

Fingerprint Sensor:

A fingerprint sensor is an electronic device that uses the unique patterns of a person's fingerprint to authenticate the identity of the person. When interfaced with microcontrollers like the Arduino Uno, it allows for safe and convenient user identification for use in various applications.Mounting a fingerprint sensor on an Arduino involves connecting the sensor's communication pins (typically TX and RX) to Arduino digital pins. Libraries such as the Adafruit Fingerprint Sensor Library can be used to assist with communication and management. The configuration offers the option to develop custom biometric authentication systems with certain project specifications.

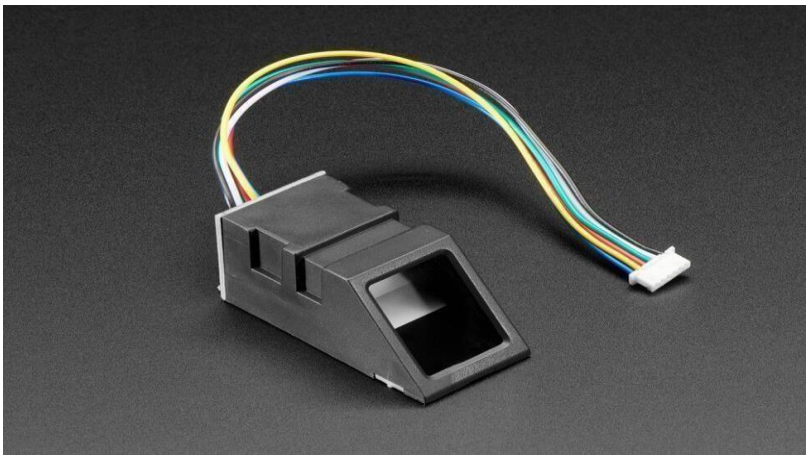

Figure 3.2: Fingerprint sensor module

Arduino UNO:

The **Arduino Uno** is an open-source microcontroller board built around the **ATmega328P** microcontroller. It's widely used in electronics projects due to its simplicity, affordability, and robust community support. The board features 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. These components provide everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Figure 3.3:Arduino UNO

Buzzer:

A buzzer is a device that emits sound when electricity flows through it. It is typically used in circuits to produce alarms or sound signals. Buzzers come in two varieties: active and passive, and they are compact and simple to use.

Figure 3.4: Buzzer

Breadboard:

A breadboard is a plastic board with lots of small holes. You can put electronic parts and wires into these holes to build a circuit without using solder. Inside, the holes are connected by metal strips so electricity can flow between parts.

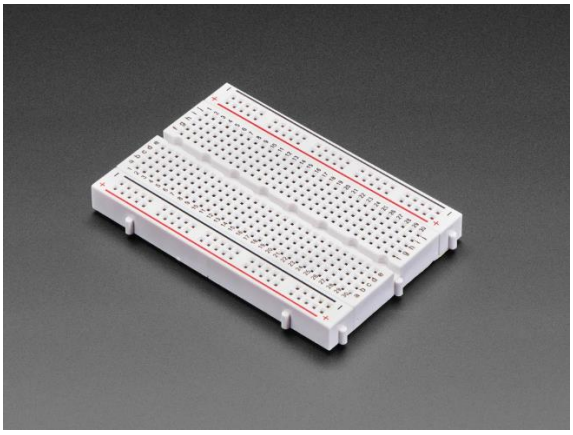

Figure 3.5:Breadboard

Jumper wires:

Jumper wires are simple wires used to connect different parts on a breadboard or between components. They have metal pins or connectors at both ends to fit easily into the breadboard holes or Arduino pins.

These wires help carry electricity from one place to another in a circuit. They come in different colors and lengths to keep the circuit neat and easy to understand. Jumper wires make building and testing circuits faster and simpler.

Figure 3.6: Jumper wires

3.3 Circuit Design:

How the System Works – In Simple Steps

Here's how the system behaves when it runs:

Startup:

Tells the user to place their finger on the scanner.

Fingerprint Scan:

The voter puts their finger on the fingerprint sensor. If the fingerprint matches a saved one, access is granted.

Vote Casting:

The voter presses one of the buttons to choose a candidate.

Confirmation:

Once a button is pressed, the system gives a beep using the buzzer.

Reset:

The system goes back to the first screen and is ready for the next person to vote.

3.4 Behind the Scenes Logic

The fingerprint sensor is connected to the Arduino UNO. The sensor has memory where some fingerprints are saved already. When someone puts their finger on it, the sensor checks if the fingerprint matches any stored one. If it matches, the system lets the person vote. After that, when a button is pressed, the system reads which button it is and gives a reply. The buzzer makes a sound to confirm the action.

3.5 Why We Made It This Way

- Simple Interface: We didn't want a complicated UI. Just a few buttons, easy enough for anyone to use.
- Secure Voting: Using fingerprints ensures that each person can only vote once.
- Budget-Friendly: We picked cheap and easily available components since it's a college project.
- Easy to Modify: The breadboard setup allows us to fix wiring or test changes quickly.
- Upgradeable: In the future, this system can be upgraded with online Aadhaar database verification or even result uploading via Wi-Fi.

Chapter 4
Circuit Diagram and Explanation

In this chapter, we are showing how all the parts in the project are connected. If the wiring is not done properly, the system won't work. So, we planned the connections carefully.

We used a breadboard, jumper wires, and Arduino UNO, which helped us to make the full setup without doing any soldering. It was easy to connect and test everything this way.

4.1 Circuit Overview

In this chapter, we are showing how all the parts in our project are connected to each other. If the wires are not connected properly, the project won't work. So, we made the connections carefully.

We used a breadboard, jumper wires, and Arduino UNO. This helped us to build the circuit without any soldering. It was simple to connect parts and test the system like this.

4.2 Connections of Components

Below is how we connected each part:

- Fingerprint Sensor:

  VCC → 3.3V or 5V (depending on the sensor)

  GND → GND
  TX → D2 (GPIO4)
  RX → D1 (GPIO5)

  This is used to scan and match the voter's fingerprint.

- Push Buttons:

  One pin of each button → GND (through a resistor)

  Other pin → D5, D6, D7 (GPIO14, GPIO12, GPIO13)

  Each button is for voting different candidates.

- Buzzer:

  Positive (+) → D8

  Negative (–)→ GND

- Power Supply:

  We used USB from laptop or power bank to give power to Arduino UNO. All other parts take power from Arduino UNO only.

4.3 Circuit Diagram:

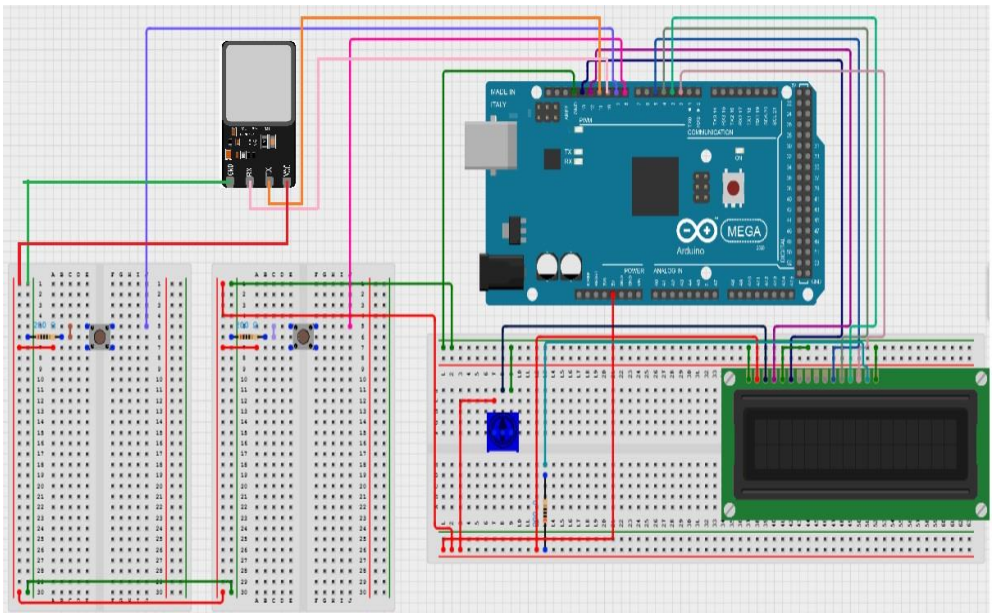

Figure 4.1: Schematic circuit diagram of voting system

4.4: How the Circuit Works:

The fingerprint sensor waits for a finger to be placed. When a valid fingerprint is detected and the button is pressed, the vote is recorded, and the buzzer gives a short beep as confirmation. After that, the system resets and gets ready for the next voter.

Why We Used Breadboard and Jumper Wires

We didn't use soldering because the breadboard is easier for testing and changes. Jumper wires made it easy to fix mistakes or try different pin connections without damaging anything.

4.5 Problems Faced During Wiring

Some jumper wires were loose and kept disconnecting.

The fingerprint sensor worked only when powered with 5V, not 3.3V.

We solved these problems step by step by checking connections and datasheets properly.

4.6 Final Output

Once all wiring was done properly, the system started working. The fingerprint matched correctly, button input was taken properly, and buzzer gave correct feedback. The complete circuit was working as expected without any error.

Chapter 5
Software Implementation

In this chapter, we are explaining how we wrote the code for our fingerprint-based voting system.

The hardware was only half of the work — the main thing that controls everything is the software. We used Arduino IDE to write and upload the code to the Arduino UNO board.

5.1 Why We Used Arduino IDE

Arduino IDE is very easy to use, even for beginners. It supports Arduino UNO and other boards, and it has many ready-made libraries that helped us a lot. We just had to select the board and COM port, and then upload the program using a USB cable.

5.2 Libraries Used:

To make different components work, we used some libraries:

Adafruit Fingerprint Library – for working with the fingerprint sensor.

SoftwareSerial.h – for serial communication between fingerprint sensor and Arduino UNO. These libraries made the coding easier and saved a lot of time.

5.3 Structure of the Code:

Like most Arduino programs, our code has two main parts:

- setup()

  This part runs only once when the system starts. We used it to initialize fingerprint sensor, and define the input/output pins for buttons and buzzer.

- loop()

  This part keeps running in a loop. It keeps checking if a finger is placed on the sensor. If the fingerprint matches, then it allows the user to vote by pressing a button.

5.4 Basic Working Logic

Here's the simple working of the code: When the user puts their finger on the sensor, it checks if the fingerprint matches a saved ID. If it matches, the voting buttons get activated. The user can then press any one button to vote. After that, the buzzer beeps to confirm the vote, and the system resets after a few seconds.

How We Handled the Voting

Each push button is connected to a different GPIO pin. When the fingerprint is verified, we read the button pins. Whichever button is pressed, we store the vote (can be stored in variables or EEPROM for demo). After voting, we disable the buttons to avoid multiple presses.

5.5 Problems Faced in Coding

Fingerprint sensor was slow sometimes. We fixed it by checking the power and adding delay. Button bouncing caused double voting, so we added small delays to handle that.

5.6 Final Testing of the Code

After writing and uploading the code, we tested the system. It was working properly — fingerprint matched fast, buzzer gave proper beep, and the buttons worked without any errors. The system was running smoothly and was ready for demo.

PROGRAM:

For enrolling:

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

SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup() {
  Serial.begin(9600);
  while (!Serial);  // for Leonardo/Micro/Zero

  delay(100);
  Serial.println("Adafruit Fingerprint sensor enrollment");

  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
}

void loop() {
  Serial.println("Ready to enroll a fingerprint!");
  Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger as...");
  while (Serial.available() == 0);
  id = Serial.parseInt();
  if (id == 0) return;

  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (!getFingerprintEnroll());
}

uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
```

```
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");
      break;
  }
  delay(100);
}

// Convert image to characteristics and store in slot 1
p = finger.image2Tz(1);
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;
}
```

```
Serial.println("Remove finger");
delay(2000);
while (finger.getImage() != FINGERPRINT_NOFINGER);

Serial.println("Place the same finger again");
p = -1;
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");
      break;
  }
  delay(100);
}

// Convert image to characteristics and store in slot 2
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;
    }

    // Create model
    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;
    }

    // Store model in flash memory
    p = finger.storeModel(id);
    if (p == FINGERPRINT_OK) {
      Serial.println("Stored!");
    } 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;
    }
  return true;
}
```

For verification & voting:

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

SoftwareSerial mySerial(2, 3);

const int buttonPin1 = 4;
const int buttonPin2 = 5;
const int buttonPin3 = 6;

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(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);

  Serial.begin(9600);
  while (!Serial);
  delay(100);
  Serial.println("Electronic Voting System Initialized");

  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Fingerprint sensor detected.");
  } else {
    Serial.println("Fingerprint sensor not found. Check wiring.");
    while (1) { delay(1); }
  }

  finger.getTemplateCount();
  Serial.print("Sensor contains ");
  Serial.print(finger.templateCount);
  Serial.println(" templates");
```

```
  Serial.println("Waiting for valid finger...");
}

void loop() {
  vote_taken = 0;
  Serial.println("\nPlease place your finger...");
  delay(1000);

  id = getFingerprintIDez();
  if (id > 0) {
    Serial.print("Voter ID: ");
    Serial.println(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";
      }

      Serial.print("Winner party: ");
      Serial.println(winner_name);
      while (1); // Halt after result
    }

    if (previous_voter_id != id) {
      do {
        Serial.println("Give Your Vote - Press a Button (1-BJP, 2-NCP, 3-Congress)");
        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++;
          vote_taken = 1;
```

```
      } else if (buttonState2 == HIGH) {
        party_2_count++;
        vote_taken = 1;
      } else if (buttonState3 == HIGH) {
        party_3_count++;
        vote_taken = 1;
      }

      if (vote_taken == 1) {
        Serial.println("Thanks for your vote.");
      }

    } while (vote_taken == 0);
  } else {
    Serial.println("Duplicate vote attempt detected!");
  }
 }
}

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;

 Serial.print("Found ID #");
 Serial.print(finger.fingerID);
 Serial.print(" with confidence of ");
 Serial.println(finger.confidence);
 return finger.fingerID;
}
```

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

Conclusion:

In this project, we made a fingerprint-based voting system that works with Aadhar. The system uses a fingerprint sensor to check the identity of the voter. If the fingerprint matches, the person is allowed to vote. This helps to stop fake voting and ensures that only the correct person can vote.

We used Arduino UNO, buzzer, push buttons, and jumper wires to build the project. The buzzer gives sound feedback. Push buttons are used to select the vote. The system is simple, low-cost, and can be useful in real elections if developed further. It makes voting safer and faster by using fingerprint-based verification.

FUTURE SCOPE:

This project is just a basic model. In the future, we can improve it in many ways: Connect it with the real Aadhar database to check real identity online.Store voting data safely in cloud storage.

- Add a mobile app to show voter details and confirmation
- Use face recognition along with fingerprint for security
- Use blockchain to make votes more secure
- Add offline features in case the internet is not working
- Improve design and cover to make it easy to use in villages and cities

Practical Applications

1. **National and Local Elections:**

- Prevents multiple voting and voter fraud by verifying identity.

2. **Corporate or Union Elections:**

- Ensures fair internal elections and secure decision-making.

3. **School and University Elections:**

- Limits voting to eligible students and avoids duplicates.

4. **Remote or Conflict Areas:**

- Enables secure, portable voting where identity verification is difficult.

5. **Digital and Blockchain Voting:**

- Adds biometric authentication for secure online voting.

ACKNOWLEDGEMENT

This work has been in part supported by the provision of a student database and input video of students in the classroom by the Department of Electronics and Communication Engineering at Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India.

APPENDIX-1

Project Outcomes and Course Outcomes Mapping

Course Outcomes (Major Projects)	Formulate hypothesis for the problem statement with sound technical knowledge from selected project domain. (CO1)	Design Engineering Solution to the problem statement with systematic approach. (CO2)	Analyse and develop an efficient solution for implementation of the project. (CO3)	Apply the theoretical concepts while providing solution to the problem statement with teamwork and multidisciplinary approach. (CO4)	Demonstrate professionalism with ethics while preparing and presenting the project work. (CO5)
Project Outcomes					

Project Outcomes -Course Outcomes- Program Outcomes (GR20) Mapping (Take guides help)				
S.No	**Project Outcomes**	**CO**	**PO/ PSO**	**Blooms Level**
1		CO1		
2		CO2		
3		CO3		
4		CO4		
5		CO5		

Sample

S.No	Project Outcome/Objectives	CO	PO/ PSO	Blooms Level
1	The advantages of RFID technology were identified and evaluated in order to design a solution to solve the existing problem.	CO1	PO1,PO2,PO4, PSO1, PSO2	Understand, Analyse, Evaluate
2	To tackle the identified problem, relevant software tools were chosen and tested.	CO2	PO3,PO4,PO5, PSO1,	Analyse,Create
3	Designed a code to control the traffic light and detect an accident.	CO3	PO3,PO5,PO8,PO,PO10	Apply,Analyse, Create
4	Teamwork, technical knowledge, general organisational abilities, and communication skills wee	CO4	PO3,PO5,PO8,PO,PO10, PSO1	Create
5	Demonstrated teamwork,technical knowledge,roganizational and communication skills by providing a solution for the existing problem.	CO5	PO9,PO11,PO12, PSO2	Create, Apply, Evaluate

Program Outcomes (GR22):

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, Engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Appendix-II

Plag report (done at GRIET)

Appendix- III

Full Paper (Published or accepted)

Receipt of Payment

Review Comments

Acceptance letter

Proceedings if published