Mini-Project Report On

SecureVoteX (A Secured Fingerprint Based Voting System)

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Aadhila Rahman (U2003001)
Aldrin Saji (U2003022)
Avelin Teresa (U2003052)
Divya R Nair (U2003071)

Under the guidance of
Dr. Saritha S

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***"SecureVoteX (A Secured Fingerprint Based Voting System)"*** *is a bonafide work done by* ***Ms. Aadhila Rahman(U2003001), Mr. Aldrin Saji (U2003022), Ms. Avelin Teresa (U2003052), Ms. Divya R Nair (U2003071)****, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2022-2023.*

Dr. Preetha K. G.	**Dr. Sminu Izudheen.**	**Dr. Saritha S**
Head of Department	Mini-Project Coordinator	Mini-Project Guide
Dept. of CSE	Professor	Assoc. Professor
RSET	Dept. of CSE	Dept. of CSE
	RSET	RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards **Dr. P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K. G.**, Head of Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini-project, ”SecureVoteX”.

We are highly indebted to our mini-project coordinators, **Dr. Sminu Izudheen**, Professor, Department of Computer Science and Engineering, **Dr. Renu Mary Daniel**, Assistant Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our mini-project guide **Dr. Saritha S**, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Aadhila Rahman

Aldrin Saji

Avelin Teresa

Divya R Nair

ABSTRACT

'SecureVoteX' is a fingerprint-based voting system designed to enhance the integrity and efficiency of the electoral process. The project aims to address the challenges of voter fraud, identity theft, and inaccurate vote counting in traditional voting systems. The primary objective is to provide a secure, user-friendly, and transparent voting platform. Through fingerprint recognition, only eligible voters can participate, eliminating the risk of fraud and multiple voting. The methodology involves utilizing a fingerprint sensor connected to a PC via a USB to TTL converter. Python's PyFingerprint library with Tkinter GUI enables voter authentication, candidate management, and encrypted vote counting. By incorporating biometric technology and streamlining voter authentication and ensuring accurate vote counting, 'SecureVoteX' ensures accurate voter identification, prevents multiple voting, and fosters trust in democratic elections.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Authentication

Biometrics is the statistical analysis for measuring biological data. It refers to technologies that calculate and find human body characteristics, such as DNA, fingerprint, retina, irises, voice or any type of noisy patterns, facial patterns and hand gesture measure, for authentication and verification purposes.

In our project, we have used fingerprint for the purpose of voter identification and authentication. Fingerprints are unique patterns found on human fingertips, formed during fetal development and unchanged throughout life. They serve as a reliable method of identification due to their individuality. No two individuals, including identical twins, have the same fingerprint patterns. The patterns include arches, loops, and whorls, each with characteristic ridges and minutiae points, creating a highly distinct fingerprint for each person. Fingerprint recognition technology is widely used for identification and security purposes. The probability of two individuals having the same fingerprints is extremely low, making them an ideal biometric identifier. Their uniqueness and permanence make them invaluable in forensic investigations and various applications, from law enforcement to modern technology.

A database is developed that stores fingerprint of every individual in the constituency. So, It checks the illegal and the repetition of votes. Hence the elections would be conducted fair and free from any type of rigging.

1.1.2 Security

Securing the database is crucial to safeguard voters data on their choices. Ensuring privacy and preventing unauthorized access is vital to maintain voter trust and uphold the integrity of the democratic process. Implementing robust encryption and access controls is essential to protect sensitive information about voters and their voting preferences.

SHA-256 is employed in our voting system not for encryption but as a cryptographic hash function to protect voter privacy, maintain data integrity, and provide anonymity for the candidate selection. Utilizing SHA-256 ensures a secure and trusted voting process that upholds the principles of democratic elections.

1.2 Existing System

Existing models of voting include paper-based voting, electronic voting machines (EVMs), internet voting, and postal voting. However, each model has its own set of challenges and potential problems:

1. Paper-Based Voting: - Challenges: Manual counting can lead to human errors and delays in results. Ballot tampering and fraud are possible. Accessibility can be an issue for people with disabilities or language barriers.

2. Electronic Voting Machines (EVMs): - Challenges: Security concerns, as EVMs can be susceptible to hacking or manipulation. Lack of transparency and auditability. Technical glitches or malfunctions can affect the accuracy of voting.

3. Internet Voting: - Challenges: Security risks and vulnerabilities, including hacking and unauthorized access. Difficulties in ensuring voter privacy and preventing coercion. Digital divide and accessibility barriers for individuals without internet access or technological literacy.

4. Postal Voting: - Challenges: Potential delays in ballot delivery and return, impacting timely vote counting. Issues with verifying voter identity and preventing fraud. Limited transparency and difficulties in ensuring ballot secrecy.

Overall, the main problems that can arise with these voting models include integrity and security concerns, lack of transparency, potential for fraud, accessibility barriers, and technological challenges. It's important to address these issues to maintain the credibility and fairness of the electoral process.

1.3 Problem Statement

The problem is to design and implement a fingerprint-based voting system that addresses the limitations and challenges associated with traditional voting methods like electoral fraud like impersonating, multiple voting, long queue at the poll center, Security concerns,bribery, Complex election procedures and verifications, Lack of transparency

The aim is to develop a secure and reliable fingerprint-based voting system called "SecurevoteX" that utilizes fingerprint recognition technology to authenticate voters and ensure the integrity of the voting process by encrypting the voting details at time of voting.

In our research, we have used fingerprints for the purpose of voter identification and authentication. As each and every individual has unique fingerprint patterns, it helps to attain maximum accuracy.

A database is developed that stores fingerprints of every individual in the constituency. So, It checks the illegal and the repetition of votes.Furthermore all the data related to vote counting is encrypted and stored in a database which is only accessible to a single admin (Election commissioner) to ensure voters privacy. Hence the elections would be conducted fair and free from any type of rigging.

1.4 Objectives

- **Design and Develop a Fingerprint Recognition System**- Robust, accurate voter Fingerprint Recognition System, ensuring high accuracy, preventing unauthorized access or manipulation.
- **Implement Secure Data Management** - Create secure database, use encryption and privacy measures to protect voter fingerprint data and information, ensuring compliance.
- **Voter Authentication** - Accurate fingerprint matching mechanism authenticates voters before allowing them to cast their vote securely.

- **Prevent Duplicate Voting** - Cross-reference captured fingerprints with database to prevent duplicate voting, ensuring each eligible voter casts only one vote.

- **Secure and Tamper-Proof Voting Process**- Create tamper-proof voting process with strict measures to prevent tampering, unauthorized access, and manipulation of votes or data.

- **User-Friendly Interface**- Develop user-friendly interface for voters and officials, ensuring easy navigation and seamless voting experience with simplified management.

- **Transparency and Auditability**- Incorporate transparency and auditability with activity logs, audit trail, and independent verification, ensuring system integrity.

1.5 Scope

The scope of fingerprint voting aims to revolutionize the electoral landscape, fostering wider voter participation, and advancing e-governance initiatives. Through this innovative solution, we aspire to uphold the essence of democracy, while adapting to the evolving needs of the electorate and leveraging technology to create a more inclusive and secure voting process.

The primary objective of this scope is to enhance the integrity and efficiency of the electoral process while safeguarding voters' data and choices. By adopting fingerprint voting, we aim to eliminate electoral fraud and ensure transparency, instilling trust and confidence in the democratic system. The integration of robust encryption and access controls will guarantee the privacy and security of voter information.

In the context of college elections, fingerprint voting can streamline the process of selecting the college council. Students' fingerprints will serve as unique identifiers, preventing any attempts at multiple voting or impersonation. This approach significantly reduces the time taken for vote counting, ensuring quicker and more accurate results. The use of encryption further fortifies the system, deterring any potential malpractices and ensuring that the electoral process remains fair and unbiased.

Beyond college elections, the scope also extends to corporate elections, where fingerprint voting can be employed for board member selections and decision-making processes.

The same secure and efficient principles apply to national elections, referendums, and plebiscites, ensuring a trustworthy electoral system that caters to all citizens, including absentee voters.

Chapter 2

Literature Review

2.1 Electronic Voting Machines

Electronic Voting Machine (also known as EVM) is voting using electronic means to either aid or take care of the chores of casting and counting votes.

An EVM is designed with two units: the control unit and the balloting unit. These units are joined together by a cable. The control unit of the EVM is kept with the presiding officer or the polling officer. The balloting unit is kept within the voting compartment for electors to cast their votes. This is done to ensure that the polling officer verifies your identity. With the EVM, instead of issuing a ballot paper, the polling officer will press the Ballot Button which enables the voter to cast their vote. A list of candidates names and/or symbols will be available on the machine with a blue button next to it. The voter can press the button next to the candidate's name they wish to vote for.

The voting is mainly carried out in steps as below:

1. Preparation: Before the voting begins, the EVM is set up and verified by election officials to ensure its proper functioning. A mock poll is often conducted to check if all buttons on the Balloting Unit work correctly and to familiarize voters with the voting process.

2. Voter Authentication: On the day of the election, voters arrive at their designated polling stations and present their voter ID cards to election officials. After verification, they are allowed to proceed to the voting area.

3. Casting Votes: In the voting area, voters enter the voting compartment one by one. Inside, they find the Balloting Unit, which displays the list of candidates and their respective party symbols. Voters choose their preferred candidate by pressing the button next to the candidate's name and symbol.

4. Indelible Ink: After casting their votes, voters are marked with indelible ink on their fingers to prevent them from voting again.

5. Control Unit and Data Storage: The Control Unit of the EVM is responsible for recording and storing the votes. It remains with the election officials and is connected to the Balloting Unit during the polling process.

6. End of Voting: At the end of the polling day, the election officials stop the voting process and disconnect the Balloting Unit from the Control Unit. The votes are then secured inside the Control Unit, and the Balloting Unit is sealed.

7. Vote Counting: After polling is complete, the Control Unit is taken to a centralized location where votes from various EVMs are counted using a software application. The data stored in the Control Unit is tallied to determine the final results.

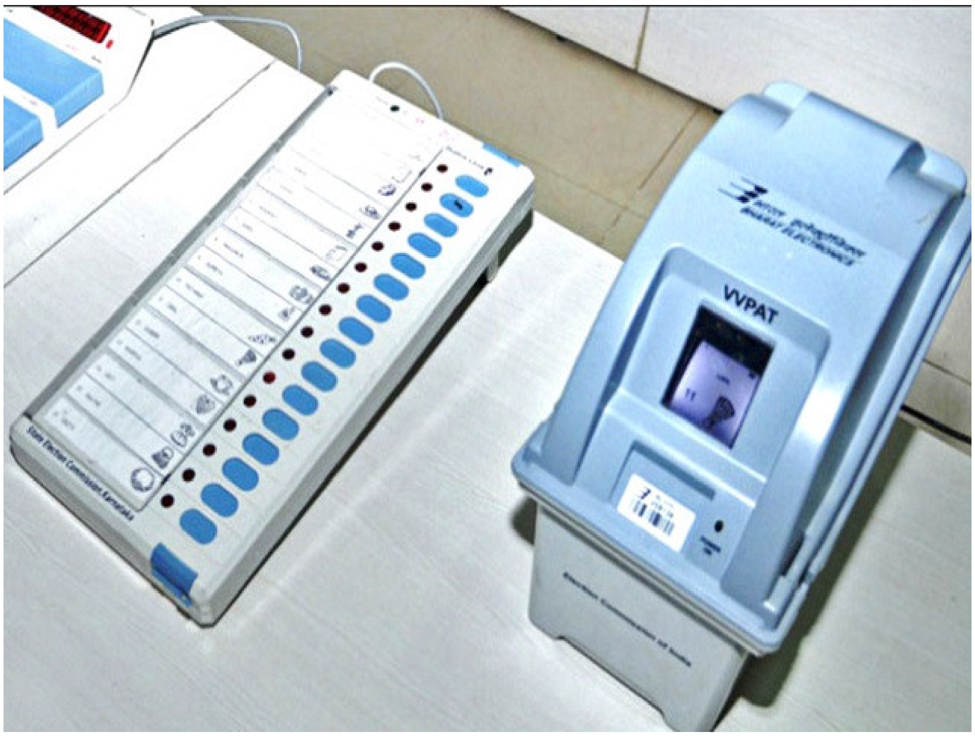


Figure 2.1: Electronic Voting Machine

However, like any technology, EVMs have many drawbacks:

1. Volume of Ballots: In elections with a large number of voters, the sheer volume of paper ballots can be overwhelming, leading to time-consuming sorting, stacking, and counting processes.

2. Lack of Voter Confidence: One significant drawback is the lack of confidence some voters have in the integrity and security of EVMs. There have been concerns and allegations regarding potential tampering or hacking, which could undermine trust in the electoral process.

3. Technical Malfunctions: EVMs can experience technical glitches or malfunctions during elections, which may lead to disruptions in the voting process and cause delays.

4. Limited Accessibility: EVMs may not be suitable for all voters, especially those with disabilities or limited technological literacy, potentially excluding certain segments of the population.

5. Geographical Spread: In countries with vast geographical areas or remote locations, collecting ballot papers from various polling stations to a centralized counting location can be time-consuming and logistically challenging.

6. Human Error: Manual counting is prone to human errors, such as miscounts or misinterpretation of voter intent, which may necessitate recounting and further delay the process.

7. Data Security Concerns: As EVMs store data electronically, there are concerns about data security and the possibility of unauthorized access or manipulation.

8. Disputed Ballots: Some ballot papers may be disputed due to unclear markings, over-voting, or other issues, requiring careful examination and resolution, which can further delay the process.

2.2 Paper ballot voting

Ballot voting, also known as paper ballot voting or traditional voting, is a method of casting and counting votes using physical paper ballots. In this system, voters are provided with a paper ballot on which they mark their choices by filling in circles or boxes next to the candidate or option of their preference. The marked paper ballots are then collected and counted manually by election officials.

Figure 2.2: Ballot

Drawbacks of paper-based Ballot voting are, it can be time-consuming, especially in elections with a large number of voters. Manual counting is also susceptible to human errors, such as miscounts or misinterpretation of voter intent, which may lead to disputes and recount requests. Additionally, paper-based voting requires significant resources for printing and handling of paper ballots, and it may not be feasible for absentee voters or those with physical disabilities.(write in points and little more details)

2.3 Internet Voting

Internet voting, also known as online voting or e-voting, is a method that allows eligible voters to cast their ballots electronically using the internet. In internet voting systems, voters are provided with secure online platforms where they can access their ballots, make their choices, and submit their votes digitally. Internet voting offers several potential benefits, including convenience, accessibility for remote or overseas voters, and potentially faster vote counting.

Disadvantages of this model are,

1. Security Concerns: Internet voting raises significant security challenges, including

the risk of hacking, voter impersonation, and vote manipulation.

2. Voter Authentication: Ensuring accurate voter authentication and preventing multiple voting are critical issues in internet voting systems.
3. Digital Divide: Internet voting may disenfranchise voters without access to reliable internet services or those with limited technological literacy.
4. Privacy Concerns: Maintaining voter privacy and ensuring that votes remain anonymous can be complex in online voting systems.
5. Auditability: The lack of a physical paper trail in most internet voting systems makes it challenging to verify and audit election results.
6. Technical Issues: Technical glitches or system failures during voting could disrupt the process and lead to voter frustration.

Due to the complexities and security concerns involved, the implementation of internet voting varies among countries. Some nations have experimented with limited internet voting for certain groups, while others have yet to adopt it on a widespread scale. Striking a balance between convenience, accessibility, and robust security remains a central challenge in adopting internet voting for broader electoral use.

2.4 Postal voting

Postal voting, also known as mail-in voting or absentee voting, is a voting method that allows eligible voters to cast their ballots by mail. In this process, voters request a postal ballot from election authorities and receive the ballot papers through the mail. They mark their choices on the ballot and then return it by mail or in person to the designated election office.

there are some drawbacks which are,

1. Potential for Fraud: Postal voting can be susceptible to fraud, such as coercion or tampering of ballots during transit.
2. Processing Time: Postal voting may take longer to process, especially during high voter turnout, which can delay the announcement of results.

3. Postal Delays: There is a risk of postal delays, which could result in some ballots not reaching election offices in time to be counted.
4. Verification Challenges: Verifying the authenticity of postal ballots and ensuring accurate voter identification can be challenging.
5. Cost: Implementing postal voting systems can involve additional costs for printing and mailing ballots.
6. Lack of Secrecy: Voters may not have the same level of secrecy as in traditional polling booths, potentially leading to concerns about voter intimidation or coercion.

Chapter 3

System Analysis

3.1 Expected System Requirements

The system of user which is a smart phone is expected to have the following features:

- Windows platform with a version above 7.
- A storage space of approximate 100 MB
- A minimum Ram size of 4GB is required in the device.
- The system must include a USB serial port.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

The technical feasibility of the Secure Fingerprint Voting System is highly promising due to the availability of mature technologies. Fingerprint recognition provides reliable voter authentication, and SHA-256 cryptographic hashing ensures candidate ID anonymity. Secure data management practices protect sensitive information, while a user-friendly interface enhances accessibility.Well-established security measures like encryption safeguard against data breaches and tampering. The system's design allows scalability for different election sizes. Overall, the combination of established technologies, secure practices, and compatibility ensures the project's technical feasibility, delivering a trustworthy, transparent, and efficient voting system.

3.2.2 Operational Feasibility

The operations are built in a simple and easy to use manner for geriatric people. Installation of the software and connecting the R307s sensor is the only prerequisite operations to be done, here we have also made an exe application for smooth transfer of software as an app.

3.2.3 Economic Feasibility

The software can reduce the overhead of expense incurred by geriatric people in order to need to print and distribute paper ballots, this eliminates the need for manual counting and reduces the labor costs associated with hiring numerous election officials to tally votes, administrative costs and human errors related to voter registration and data management.Fingerprint voting systems can be used in multiple locations, eliminating the need to set up physical polling stations everywhere. This reduces logistics as votes can be stored digitally in database and need not carry these vote suitcases to counting location and their security cost can also be reduced and setup costs while providing greater accessibility to voters.The entire system was built with budget of Rs.1400 which includes R307s, USB to TTL converter and all wires, rest were built using free resources.

3.3 Hardware Requirements

The following are the system requirements to develop the project.

- Processor: Intel Core i3
- Hard Disk: Minimum 5GB
- RAM: Minimum 1GB
- USB TO TTL Converter
- Fingerprint Scanner r307s

3.4 Software Requirements

The following are the softwares used in the development of the project.

- Operating System: Windows
- Database: SQLite3

- Editor: Visual Studio Code
- Programming Language: Python

3.4.1 Visual Studio Code

Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft. It offers an extensive range of features, such as intelligent code completion, debugging support, version control integration, and syntax highlighting. Being lightweight and fast, it is suitable for various projects. VS Code is cross-platform, available for Windows, macOS, and Linux, and boasts a large library of extensions to enhance functionality.

With an integrated terminal and active community, Visual Studio Code has become a go-to tool for developers worldwide.

3.4.2 SQLiteStudio

SQLiteStudio, is an open-source, multi-platform SQLite database manager. It is designed to provide a user-friendly interface for managing and interacting with SQLite databases. SQLiteStudio allows users to create, modify, and execute SQL queries, as well as browse, edit, and export database contents. It supports various operating systems, including Windows, macOS, and Linux, making it a versatile tool for developers and database administrators working with SQLite databases. With its intuitive interface and useful features, SQLiteStudio is a popular choice for efficiently managing SQLite databases for various applications.

Chapter 4

Methodology

4.1 Proposed Method

- Fingerprint is used to uniquely identify the user.
- Sensor is connected to pc via usb to ttl converter
- Instructions are sent from Python tkinter GUI to sensor via serial port communication
- Multiple voting and unauthorized logins are prevented.
- Fingerprint data is read and stored in database connected to python.
- Voting details are encrypted and used for vote counting.
- Encrypted data and count is mapped to candidate names.
- Results are published in GUI.

4.1.1 Fingerprint Sensor

The R307s fingerprint sensor is a commonly used biometric sensor that provides fingerprint recognition and user authentication capabilities. As a standalone module, it incorporates various components and functionalities to accurately capture and process fingerprints for identification purposes.

R307s Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions

Figure 4.1: R307s Fingerprint Sensor

Specifications

- Supply voltage: DC 4.2   6.0V
- Supply current: Working current: 50mA (typical) Peak current: 80mA
- Fingerprint image input time: ¡0.3 seconds
- Window area: 14x18 mm
- Matching method: Comparison method (1: 1)
- Search method (1: N)
- Characteristic file: 256 bytes
- Template file: 512 bytes
- Storage capacity: 1000 pieces
- Security Level: Five (from low to high: 1,2,3,4,5)
- Fake rate (FAR): ¡0.001
- Refusal rate (FRR): ¡1.0

- Search time: ¡1.0 seconds (1: 1000 hours, mean value)
- Host interface: UART USB1.1
- Communication baud rate (UART): (9600xN) bps Where N = 1 12 (default N = 6, ie 57600bps) Working environment: Temperature: -20 ℃ - +40 ℃ Relative humidity: 40
- Storage environment: Temperature: -40 ℃ - +85 ℃ Relative humidity: ¡85

Working Principle

An optical fingerprint scanner works based on the principle of Total Internal Reflection (TIR). In an optical fingerprint scanner, a glass prism is used to facilitate TIR. Light from an LED (usually blue color) is allowed to enter through one face of the prism at a certain angle for the TIR to occur. The reflected light exits the prism through the other face where a lens and an image sensor (essentially camera) are placed.

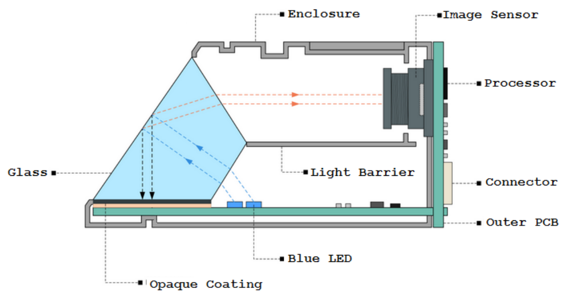


Figure 4.2: R307s Fingerprint Sensor cross-section

When there's no finger on the prism, the light will be completely reflected off from the surface, producing a plain image in the image sensor. When TIR occurs, a small amount of light leaked to the external medium and it is called the Evanescent Wave. Materials with different refractive indexes (RI) interact with the evanescent wave differently. When we touch a glass surface, only the ridges make good contact with it. The valleys remain separated from the surface by air packets. Our skin and air have different RIs and thus affect the evanescent field differently. This effect is called Frustrated Total Internal Reflection (FTIR). This effect alters the intensities of the internally reflected light and is detected by the image sensor (see this image). The image sensor data is processed to produce a high contrast image which will be the digital version of the fingerprint.

In capacitive sensors, which are more accurate and less bulky, there's no light involved. Instead, an array of capacitive sensors are arranged on the surface of the sensor and allowed to come in contact with the finger. The ridges and air packets affect the capacitive sensors differently. The data from the sensor array can be used to generate a digital image of the fingerprint.

Above is a cross-sectional diagram(see fig 4.2) to understand the construction (illustrative only, not a physically exact one). Opening the module was easy; there are four Philips screws on the back. Unscrew them and it can remove the PCB. There are two PCBs; one arranged horizontally and one vertically (shown in washed green). These PCBs are connected by solder. The four blue LEDs and the touch sense pad are on the horizontal PCB. The vertical PCB has the image sensor, the processor and connector. When inserted, the touch sense pad comes in contact with the glass block above. The image sensor is soldered and glued. Strangely, I couldn't find any lens on it. May be it doesn't need one. The enclosure has an internal barrier to separate the light from the LEDs and the light coming out of the prism. On the bottom side of the prism a black epoxy is coated which gives a high-contrast background for the fingerprint image. To access the prism, just remove the cap on the front.

PIN Function

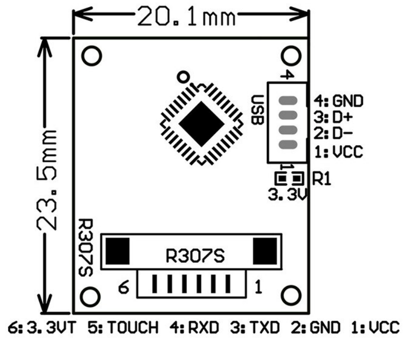


Figure 4.3: PIN Diagram(R307s)

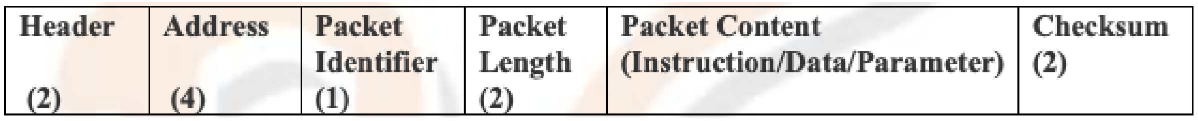
- 5V Regulated 5V DC
- GND Common Ground
- TXD Data output - Connect to MCU RX
- RXD Data Input - Connect to MCU TX
- TOUCH -Active Low output when there is touch on sensor by finger
- 3.3V -Use this wire to give 3.3V to sensor instead of 5V

MEMORY AND REGISTERS

- Notepad - This is 512 bytes of the non-volatile flash memory. It is logically divided into 16 pages with 32 bytes each. Instructions GR_WriteNotepad and GR_ReadNotepad can be used to access this memory. When writing a page, it is taken as a whole, and the contents are replaced.
- Image Buffer - Image buffer is used to store a BMP image of size 256 x 288, each pixel occupying a byte. This buffer is part of the RAM and the contents are lost when power is lost
- Character File Buffer - A character file is a processed high contrast image of a fingerprint.Two-character files from two consecutive scans are combined to form a template file which is the final version of the fingerprint that is stored in the fingerprint library (not to be confused with the Arduino library. Fingerprint library is the memory used to store up to 1000 fingerprints). The two-character file buffers are CharBuffer1 and CharBuffer2 each with size of 512 bytes.
- Fingerprint Library - This is a section of the flash memory where 1000 fingerprint templates can be stored. Templates are arranged sequentially with numbering from 0 to N-1 (The manual says 0-N) where N is the capacity of the library determined by the size of the flash memory. There are instructions to store, process and delete templates from this memory. They will be explained later.
- System Configuration Register - This is a 16-bytes long register bank containing operating parameters and status. Except the device address which takes up 4 bytes,

rest of the parameters are 2 bytes (a word) in length. The command ReadSysPara can be used to read, and command SetSysPara can be used to write this register bank.

Packet Format



Header (2)	Address (4)	Packet Identifier (1)	Packet Length (2)	Packet Content (Instruction/Data/Parameter)	Checksum (2)

Figure 4.4: Packet Format

A packet is a group of many such bytes (or frames).

- Header: This indicates the start of a packet. It has to be the fixed value 0xEF01. It is 2 bytes long and the high byte is always transferred first.
- Address: This is the 32-bit address of the scanner module. The module will accept instructions only if the address we are sending matches the address stored in the module. The default address is 0xFFFFFFFF and can be modified with SetAddr instruction.
- Packet Identifier: This determines what type of packet we're sending or receiving. It is 1 byte long
- Packet Length: This is the total length of Packet Content and Checksum in bytes. Maximum length is 256 bytes and high byte is transferred first
- Packet Content: This can be data/command/parameters etc. of varying length. The Packet Length is the value that specifies the length of the data here in bytes.
- Checksum: This is the arithmetic sum of all bytes in Packet Identifier, Packet Length and Packet Content. Overflowing bits are ignored. High byte is always transferred first.

4.1.2 Graphical User Interface

Tkinter is a standard Python library used for creating Graphical User Interfaces (GUIs). It provides a set of built-in widgets and functions that allow developers to create windows, dialogs, buttons, menus, textboxes, and other graphical elements for desktop applications. Tkinter is based on the Tk GUI toolkit, which is a cross-platform graphical user interface toolkit written in C.

FEATURES

- Cross-platform: Tkinter applications can run on different platforms, including Windows, macOS, and Linux, without modifications.
- Easy to use: Tkinter provides a simple and intuitive API, making it easy for beginners to create GUI applications in Python.
- Wide range of widgets: Tkinter offers a variety of widgets like buttons, labels, textboxes, checkboxes, radio buttons, and more, allowing developers to create interactive interfaces.
- Event-driven programming: Tkinter follows an event-driven programming paradigm, where GUI elements respond to user interactions (such as button clicks or mouse movements) through event handlers.
- Integration with Python: Tkinter is a part of the Python standard library, so no additional installations or dependencies are required to use it.
- Object-oriented: Tkinter follows an object-oriented approach, enabling the creation of reusable and modular GUI components.
- Customizable: Developers can customize the appearance and behavior of widgets by changing their attributes and options.

4.1.3 Database

SQLITE3

SQLite3 is a lightweight, serverless, self-contained, and open-source relational database management system (RDBMS). It is widely used as a local database engine in various applications due to its simplicity, ease of use, and portability. Unlike traditional database management systems, SQLite3 does not require a separate server process to be running, making it ideal for embedded systems, mobile applications, and small-scale projects.

FEATURES

- Serverless Architecture: SQLite3 is a serverless database, meaning it operates directly with the application without requiring a dedicated server process. The entire database is contained within a single file, making it easy to manage and transport.
- Self-contained: The SQLite3 library and engine are self-contained and can be linked directly into the application, eliminating the need for external dependencies or installations.
- SQL Support: SQLite3 supports a subset of SQL (Structured Query Language), allowing users to perform standard SQL operations like creating, reading, updating, and deleting data in a relational database.
- ACID Compliance: SQLite3 ensures the ACID (Atomicity, Consistency, Isolation, Durability) properties, making it reliable and robust for transactional operations.
- Small Footprint: SQLite3 is designed to have a small memory and storage footprint, making it suitable for resource-constrained environments.
- Portability: SQLite3 databases can be used across different platforms and operating systems without any modification.
- Extensible: SQLite3 supports user-defined functions, triggers, and custom collation sequences, allowing developers to extend its functionality.

Chapter 5

System Design

5.1 Architecture Diagram

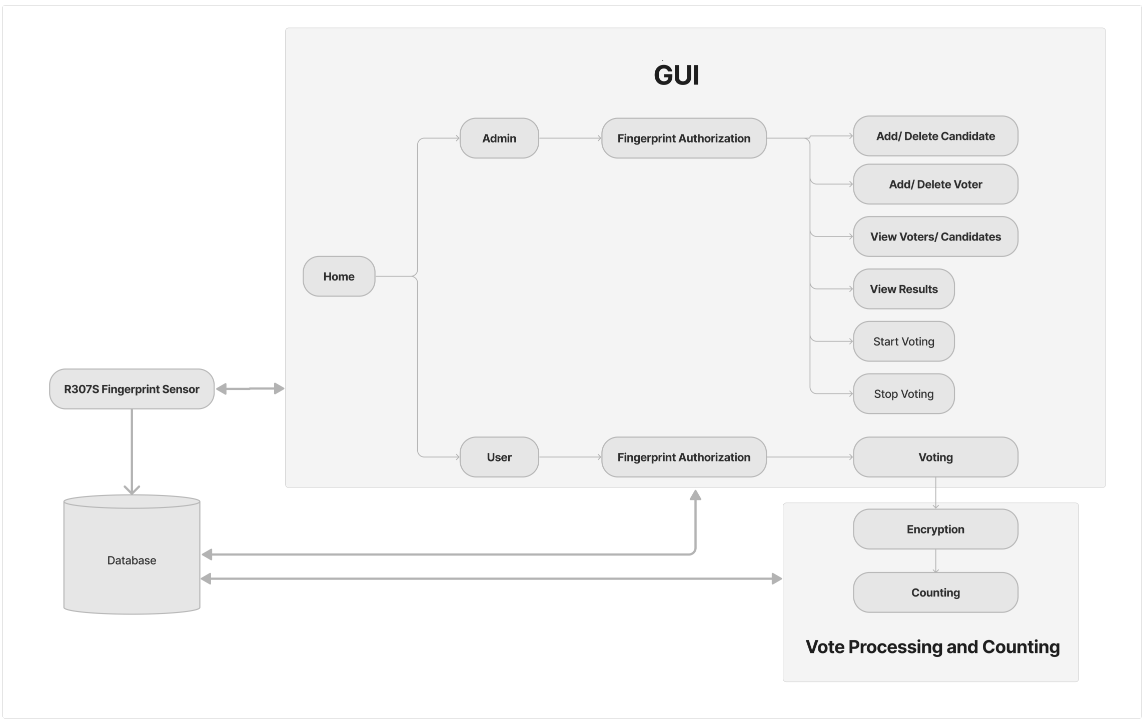


Figure 5.1: Architecture diagram

5.2 UML Diagram

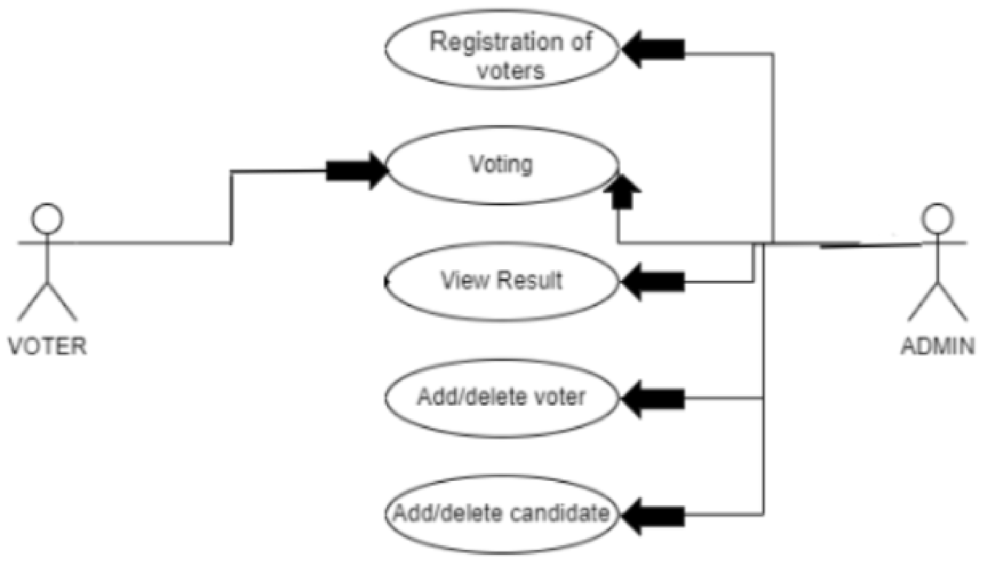


Figure 5.2: Face Recognition

Chapter 6

System Implementation

6.1 Fingerprint Sensor Interface Module:

The Fingerprint Sensor Interface Module in the fingerprint voting system is a crucial component that handles the communication with the R307S fingerprint sensor through the USB to TTL converter.Its main purpose is to provide a seamless integration between the sensor and the Python application. This module allows for functions related to enrolling, verifying, and deleting fingerprints from the sensor's memory.

6.1.1 Model Architecture

- Search for Available Ports: When the fingerprint voting system is launched, a loading window is displayed to the user. This loading window is responsible for searching for available ports on the PC to which the R307S fingerprint sensor might be connected. It scans through the available serial ports to find the one where the sensor is connected.

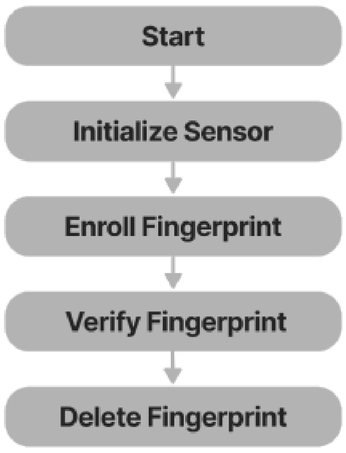
- Establish Connection: Once the available port for the fingerprint sensor is found, the loading window establishes a connection with the sensor using the pyfingerprint library. The library provides methods to interact with the sensor through the established serial connection. The loading window initializes the connection object, which can later be imported by other window programs, ensuring that the connection is accessible throughout the application's runtime.The module establishes a connection to the R307S fingerprint sensor through the USB to TTL converter. This involves configuring the communication settings, such as baud rate and data format, to enable

- Verification of Password: After establishing the connection, the loading window proceeds to verify the password for the fingerprint sensor using the verifyPassword() function provided by the pyfingerprint library. The fingerprint sensor is typically protected with a password to ensure secure access.

- Importing Connection Object: Once the connection is successfully established and the password verification is done, the loading window imports the connection object. This makes the connection accessible by other window programs within the application. The connection object can then be utilized by other modules or GUI screens to perform fingerprint-related functions such as enrollment and verification.
- Authentication: Before accessing the sensor's features, the module must authenticate itself by sending the correct password to the sensor. This ensures that only authorized applications can interact with the fingerprint sensor.
- Enroll Fingerprint: The module provides a function to enroll a new fingerprint. When called, this function guides the user through the enrollment process, capturing multiple samples of their fingerprint. These samples are then stored in the sensor's memory for future verification.
- Verify Fingerprint: The module includes a function to verify a user's fingerprint against the enrolled templates. When invoked, this function prompts the user to place their finger on the sensor. It captures the fingerprint sample and compares it with the stored templates to determine if there is a match.
- Delete Fingerprint: To manage the enrolled fingerprints, the module offers a function to delete specific fingerprints from the sensor's memory. This allows for proper administration of the fingerprint database.

The Fingerprint Sensor Interface Module serves as a crucial intermediary between the fingerprint sensor and the application. Its successful implementation ensures a secure and reliable biometric authentication process. By providing these essential functions, the module enables the application to interact seamlessly with the fingerprint sensor, facilitating user enrollment, verification, and deletion, and enhancing the overall accuracy and security of the voting system.

6.1.2 Flowchart



6.2 Database Module

The Database Module in the fingerprint voting system is responsible for managing and storing data related to candidates, voters, and system settings. It utilizes SQLite 3, a lightweight and efficient relational database management system, to store data securely on the local machine. The module provides functions to interact with the database and perform operations such as adding, deleting, and retrieving data. Additionally, it manages the encryption and integrity of sensitive voting data.

6.2.1 Model Architecture

- Initialization: The module starts by establishing a connection to the SQLite 3 database. It creates the necessary tables for storing candidate information, voter details, and system settings during the setup process.

- Candidate Table: The module contains functions to add new candidates and delete existing ones. It stores candidate data, such as their unique identifier (candidate ID), name, and image location, in the candidate table.

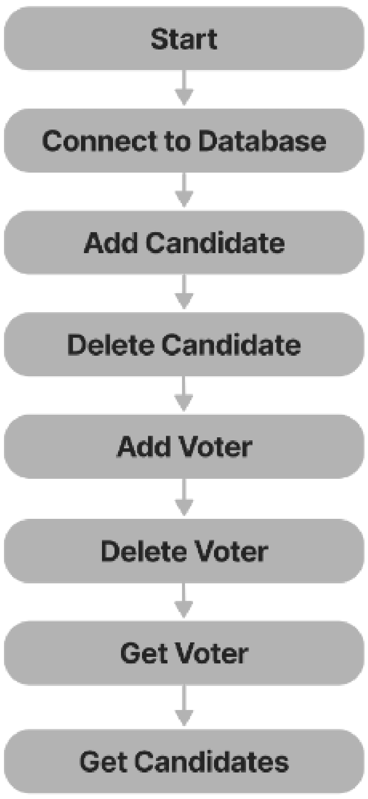
- Voter Table: The module includes functions to add new voters and delete existing ones. It stores voter data, including the unique voter ID, name, image location and fingerprint template id, in the voter table.

- Vote Recording: When a voter casts their vote, the module records the encrypted candidate ID corresponding to the chosen candidate in the database. This data is stored in a separate table, ensuring the secrecy and integrity of the votes.
- System Settings: The module manages system settings, such as whether voting is currently enabled or stopped by the administrator. It stores this information in the system settings table, with the voting enabled column having values 0 or 1, indicating if voting is active or disabled. If voting is stopped by the administrator, the module allows users to view the election results while preventing any further voting activities.
- Duplicate Voting Prevention: To prevent duplicate voting, the module maintains a text file containing the fingerprint IDs of voters who have already cast their votes. Before accepting a vote, the system checks this file to ensure that a voter has not already voted in the current session.
- Data Encryption: The module ensures data security by encrypting sensitive voting information, such as the candidate IDs, before storing them in the database . After each voting session, the module stores the encrypted data of candidate IDs who received votes in a separate table. The vote count is also updated in the well after mapping with encrypted id.

The Database Module serves as a vital component of the fingerprint voting system, facilitating secure and efficient data management. By utilizing SQLite 3, it provides a reliable and scalable database solution while ensuring the integrity and confidentiality of voting data. The module's functions enable administrators to manage candidate lists, voters, and system settings seamlessly, ensuring a smooth and transparent voting process for both administrators and voters.

6.2.2 Flowchart

.



6.3 Encryption and Vote Processing Module

The Encryption and Vote Processing Module is a critical component of the fingerprint voting system responsible for processing cast votes, ensuring data integrity, and generating accurate voting results. This module operates based on the "voting enabled" status, which indicates whether voting is currently allowed or stopped by the admin. It utilizes hash functions of SHA-256 for encrypting the candidate IDs to ensure data security and prevent tampering. Additionally, the module prevents duplicate voting by cross-referencing fingerprint IDs with a text file containing previously voted IDs.

6.3.1 Model Architecture

- Vote Processing:
  - When a voter casts a vote, the fingerprint is captured and authenticated by the Fingerprint Sensor Interface Module. The voter's fingerprint ID is then checked against the text file that stores the IDs of those who have already voted to prevent duplicate voting.

  - If the voter's fingerprint ID is not found in the file and ”voting enabled” is set to 1, the vote is accepted and processed. Otherwise, if ”voting enabled” is set to 0, the vote is rejected, and the voter is informed that voting is currently stopped.

- Encryption of Candidate IDs:
  - After a successful vote is cast, the Vote Processing, Encryption, and Counting Module encrypts the candidate ID using SHA-256 hash function. This one-way encryption ensures that the candidate IDs cannot be reverse-engineered or tampered with.
  - The encrypted candidate ID is stored in the database in a separate table specifically designed to store the encrypted data for candidates who received votes.

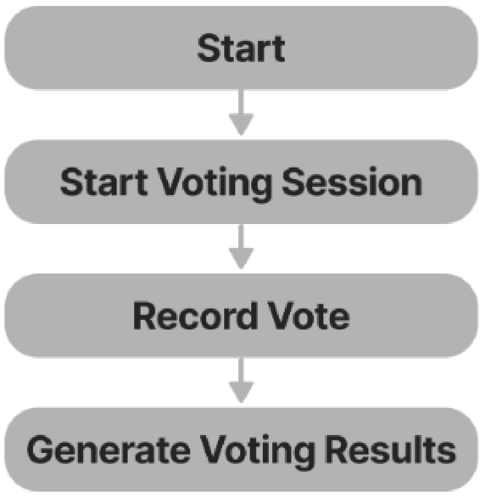
- Vote Counting:
  - The module maintains the ”vote count” for each candidate. When a vote is processed, the encrypted candidate ID is checked in the encrypted data table to determine if the candidate has already received votes.
  - If the candidate has not received any votes yet, a new entry is created with the encrypted candidate ID, and the vote count is initialized to 1.
  - If the candidate has previously received votes, the module increments the vote count for the corresponding encrypted candidate ID.

- Results and Viewing:
  - The voting system allows administrators to change the ”voting enabled” status. When voting is stopped (voting enabled = 0), only the results can be viewed, and no new votes are accepted.
  - After voting is closed, administrators can access the voting results, which are stored in the encrypted data table. The vote count for each candidate is decrypted and presented to display the final voting outcomes.

The Encryption and Vote Processing module ensures the integrity of the voting process, prevents fraudulent voting, and produces accurate voting results. By using SHA-256 encryption, the system maintains data security, while the cross-referencing of fingerprint IDs prevents duplicate voting. This module plays a crucial role in the transparency and fairness of the election process in the fingerprint voting system.

6.3.2 Flowchart

.



6.4 GUI Module

The GUI Module in the fingerprint voting system is developed using the Tkinter library in Python. It provides a user-friendly and intuitive interface for both administrators and voters to interact with the system. The GUI module is designed to accommodate functionalities for voter authentication, candidate management, voting process, and viewing voting results based on the "voting enabled" status. The GUI module also incorporates a loading window, which searches for available ports and establishes the connection to the fingerprint sensor via USB to TTL converter during the system startup. Additionally, the application is packaged into a .exe executable for ease of use and distribution.

6.4.1 Model Architecture

- Loading Window:

  - Upon launching the application, a loading window appears, indicating that the system is initializing. During this phase, the GUI module searches for available ports to establish the connection with the fingerprint sensor via the USB to TTL converter.
  - Once the appropriate port is found, the connection to the sensor is established, and the loading window progresses to the main application window.

- Main Application Window:
  - The main application window displays two buttons: "Admin" and "User," representing the two roles available in the system.
  - Upon clicking the "Admin" button, the GUI module prompts the administrator to read fingerprint for authentication before accessing admin functionalities.
  - Clicking the "User" button directs the user to the Fingerprint Authentication Screen and then to voting window after successful authentication.

- Admin Authentication Screen:
  - This screen prompts the administrator to read their fingerprint to verify their identity.
  - The GUI module validates the fingerprint against predefined admin fingerprint id stored securely in the fingerprint library, using pyfingerprint library functions.
  - Upon successful authentication, the Admin Panel Screen is displayed; otherwise, an error message is shown.

- Admin Panel Screen:
  - Once authenticated, the Admin Panel Screen provides various options for administrators to manage candidates and voters.
  - Administrators can add new candidates and voters, delete existing candidates and voters, view the list of candidates and voters, and access the voting results.

- Fingerprint Authentication Screen:

  - When a user clicks the "User" button, they are directed to the Fingerprint Authentication Screen.
  - This screen captures the user's fingerprint through the Fingerprint Sensor Interface Module and prompts them to place their finger on the sensor for verification.
  - The GUI module checks the captured fingerprint against the enrolled templates in the database and proceeds to the Voting Screen upon successful verification.

- Voting Screen:
  - The Voting Screen displays the list of candidates with their respective party affiliations.
  - The user selects their preferred candidate by clicking on the candidate's name.
  - The GUI module processes the vote through the Vote Processing, Encryption, and Counting Module.
  - If "voting enabled" is set to 0, the Voting Screen displays a message that voting is currently stopped.

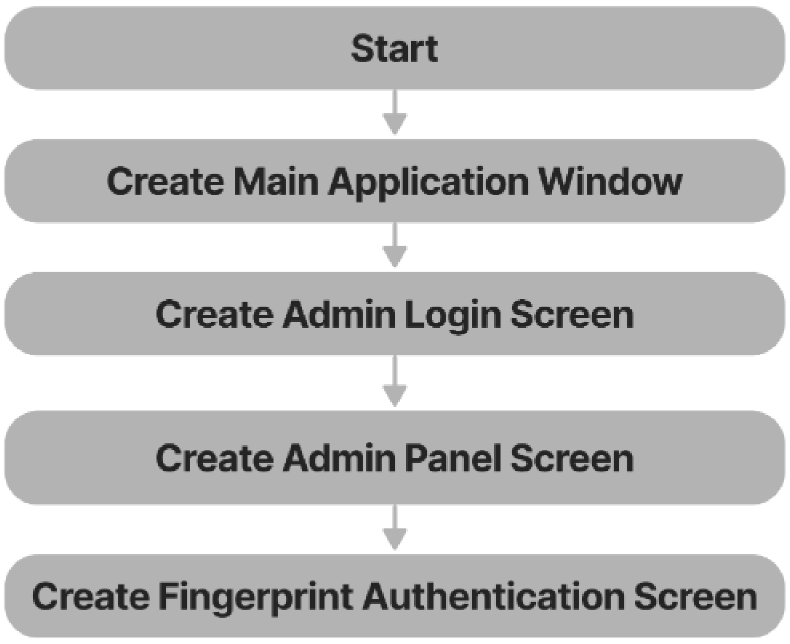
- Results Screen:
  - After voting is closed, administrators can access the voting results through the Admin Panel Screen.
  - The GUI module decrypts the encrypted data from the encrypted data table and displays the final vote count for each candidate.

- Packaging to .exe Application:
  - The entire fingerprint voting system, including the GUI module, is packaged into a standalone .exe application for easy distribution and use.
  - This allows users to run the application without the need for a Python environment or external dependencies.

The GUI module plays a vital role in providing a user-friendly and intuitive interface for both administrators and voters. It ensures a seamless experience, from the loading window to the admin functionalities, fingerprint authentication, voting process, and result display. The packaging of the application into a .exe executable enhances accessibility and ease of deployment for users.

6.4.2 Flowchart

.


Start
Create Main Application Window
Create Admin Login Screen
Create Admin Panel Screen
Create Fingerprint Authentication Screen

6.5 Main Application Module

The Main Application Module serves as the core of the fingerprint voting system, coordinating the interactions between different modules and providing the main entry point for the application. It manages the graphical user interface (GUI) by creating the main window and handling user actions, such as clicking on the admin or user buttons. This module is responsible for launching the application, showing the loading window to establish the connection with the fingerprint sensor, and initiating the appropriate functionalities based on the selected role (admin or user).

6.5.1 Model Architecture

- Application: The Main Application Module is packaged as an executable (.exe) application, making it easily accessible to users without requiring any specific development environment or Python installation. Users can simply run the .exe file to launch the fingerprint voting system.

- Loading Main Window: When the .exe application is executed, the Main Application Module first loads the main window of the application created using using the Tkinter library. This window acts as the primary interface where users can select their roles (admin or user) and access the corresponding functionalities.

- Displaying the Loading Window: After the main window is created, the Main Application Module displays the loading window. This loading window searches for available ports to establish a connection with the R307S fingerprint sensor. After loading finishes, main window is shown.

- Establishing Connection with the Fingerprint Sensor: The loading window searches for available ports and establishes a connection with the fingerprint sensor using the pyfingerprint library. Once the connection is established and the password is verified, the connection object is imported and made available to other window programs and modules within the application.

- Handling User Actions: The main window created by the Main Application Module contains two buttons: "Admin" and "User." When the user clicks on either button, the Main Application Module handles the respective user action.

- Admin Interface: If the "Admin" button is clicked, the Main Application Module displays the fingerprint authentication screen using the GUI module. Administrators can enter their login credentials to access the admin functionalities.

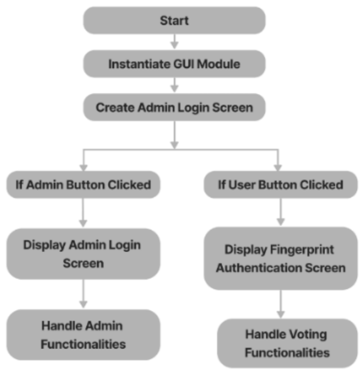
- User Interface: If the "User" button is clicked, the Main Application Module displays the fingerprint authentication screen using the GUI module. Users can authenticate themselves using their fingerprints before proceeding to the voting screen.

- Coordinating Module Interactions: The Main Application Module coordinates the interactions between various modules, such as the Fingerprint Sensor Interface Module, Database Module, Voting Module, and GUI Module. It passes information and requests between modules to perform specific operations, such as voter authentication, candidate management, and vote recording.

- Overall Control Flow: The Main Application Module is responsible for the overall control flow of the fingerprint voting system. It ensures that the necessary modules are initialized and interact seamlessly to provide a secure, efficient, and user-friendly voting experience.
- Error Handling: The Main Application Module incorporates error handling mechanisms to address unexpected scenarios or exceptions that may occur during the application's execution. Proper error messages and alerts are displayed to users when necessary.
- Closing the Application: When the user decides to exit the application, the Main Application Module handles the closing process, ensuring that all resources are properly released and the application terminates gracefully.

By acting as the central coordinator, the Main Application Module ensures the smooth functioning of the fingerprint voting system. It manages the GUI, establishes the connection with the fingerprint sensor, and orchestrates interactions between different modules to provide a reliable and user-friendly voting experience for administrators and voters alike.

6.5.2 Flowchart


Start
Instantiate GUI Module
Create Admin Login Screen
If Admin Button Clicked
If User Button Clicked
Display Admin Login Screen
Display Fingerprint Authentication Screen
Handle Admin Functionalities
Handle Voting Functionalities

Chapter 7

Testing

As the project is on a bit large scale, we always need testing to make it successful. If each component works properly in all respect and gives desired output for all kind of inputs then the project is said to be successful. So the conclusion is to make the project successful, it needs to be tested. The main testing done was System Testing in order to check whether the user requirements were being satisfied or not. The code for the new system has been written completely using visual studio with python as the coding language, along with SQLite for storing the voter and candidate details. Our system has been tested with the help of various user and application has been successfully verified from all the aspect.

Although we were able to find a few errors in the application, all these errors were analysed and hence were corrected making the application error-free. The overall flow of the application is made quite similar to the actual process.

In order to make the application error-free we have deployed a concept of multiple levels of testing before the final user acceptance testing takes place. The complete steps taken for a different level of testing are:

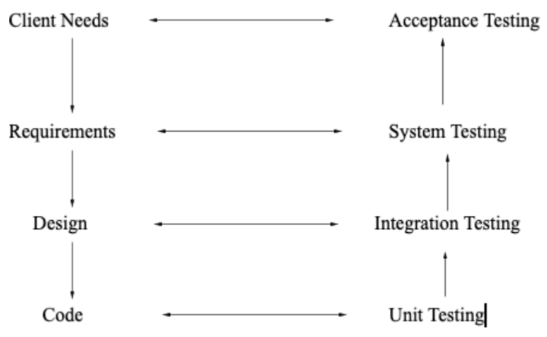


Figure 7.1: Testing

7.1 Unit Testing

In the process of Unit testing, we try to test the application by testing the units of the application or we could say the modules. The complete application is divided into different sets of Modules and each module is hence tested separately in order to scan out even the smallest possible error in the application. This is also commonly known as module testing.

7.2 Integration Testing

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for constructing the program structure while at the same time testing to uncover errors associated with the interface. The primary aim is to take Unit tested Modules and building the complete structure as a whole. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicated by the vast expense of the entire program. Thus in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

7.3 System testing

System testing is a process where the Complete and integrated software is tested for any errors. It is a type of black box testing technique. In system testing the software is tested as a whole. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

7.4 Validation Testing

After performing all the above levels of-of testing the software now has been bundled as complete package uncovering all the possible error and these errors have been corrected the final series of testing begins which is validation testing.

In the process of validation testing, we have to test the software such that it can be easily accepted by the customer. Once we have performed the validation testing either one of the two situations exists in front of us ie, the function or the performance of the application is accepted and satisfies the client need or the application deviates from the expectations of the client and is a need to debug.

7.5 Output testing

Once validation testing is over, the nest we move on to the Output testing. As we all know any system with output without in proper format is not considered as a valid system. Hence we ask the users about the required format of the output and testing the system against the given system generated output. The final results are shown on the GUI window.

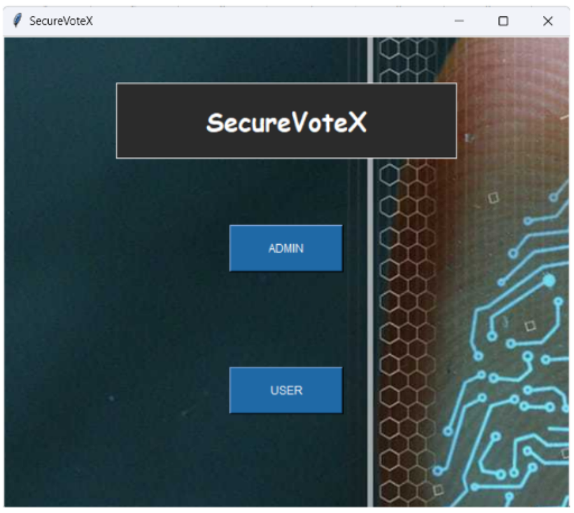
7.6 User Acceptance Testing

User acceptance of a system plays a very vital role in deciding the overall success of the system. The system under evaluation is constantly tested by the system's user for user acceptance. We constantly stay in touch with the users at the time of developing and make changes whenever required.

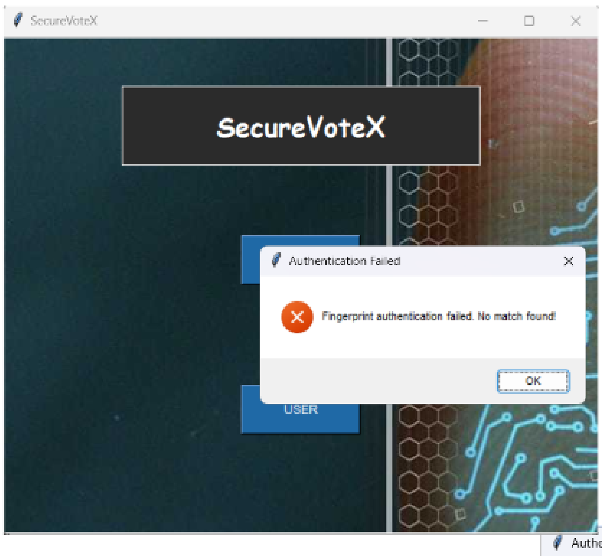
Chapter 8

Results

Main Page

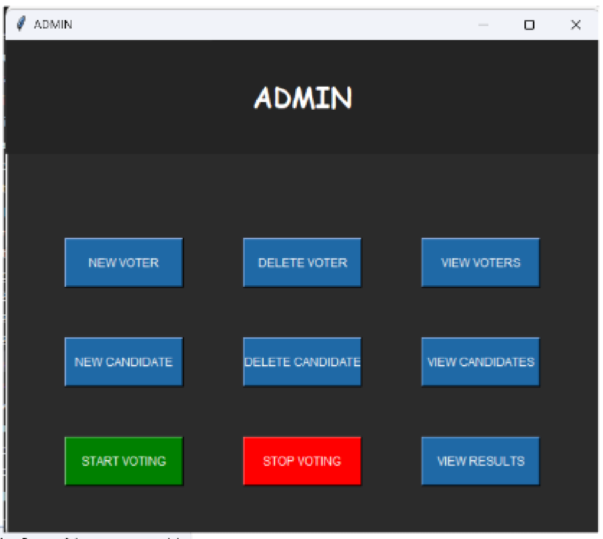


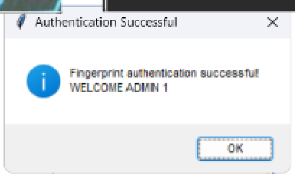
Authentication



Fingerprint authentication failed. No match found!

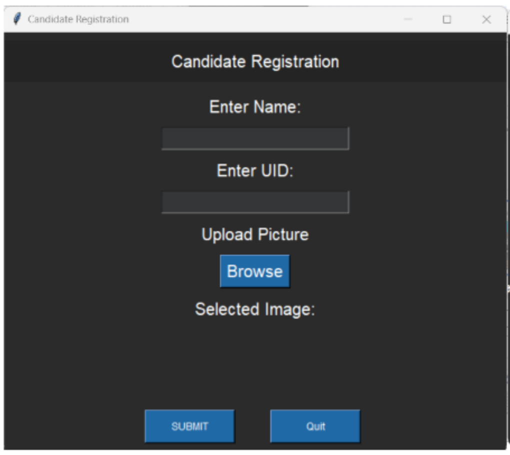
Admin Page



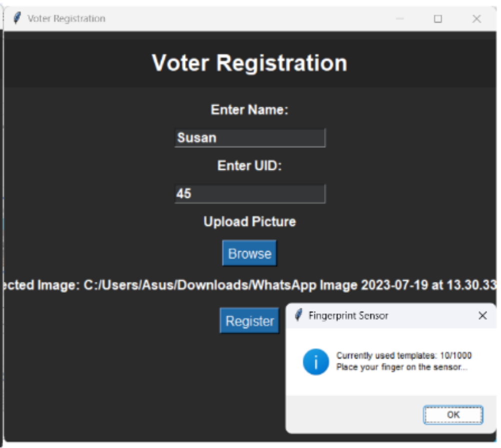


Fingerprint authentication successful
WELCOME ADMIN 1

Add Candidate



Add Voter



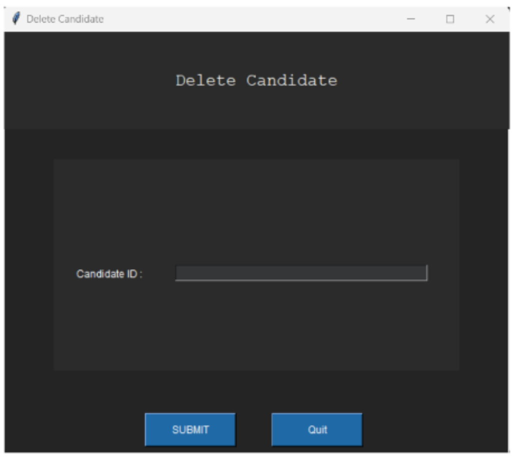
Susan

45

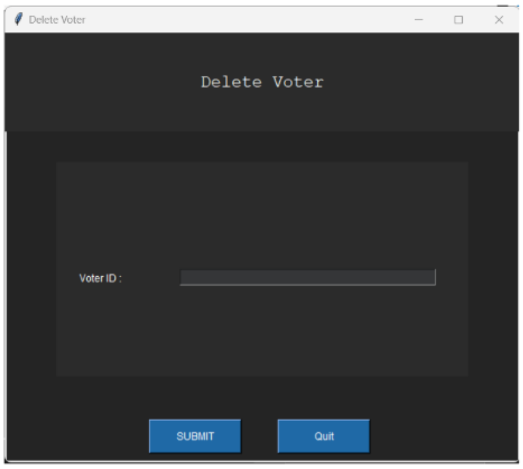
ected Image: C:/Users/Asus/Downloads/WhatsApp Image 2023-07-19 at 13.30.33

Currently used templates: 10/1000
Place your finger on the sensor...

Delete Candidate

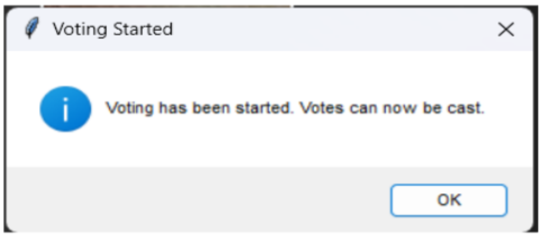


Delete Voter



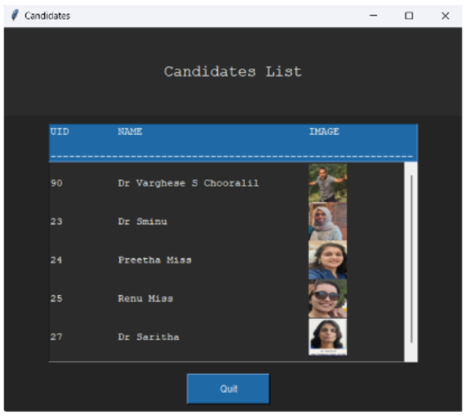
Vote control of admin

Voting has been stopped. No further votes will be accepted.



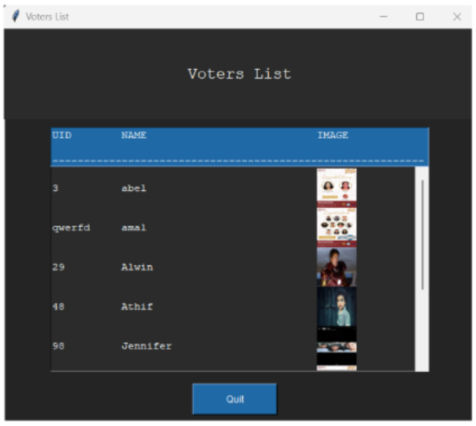
Voting has been started. Votes can now be cast.

Voters and Candidates List



Candidates List

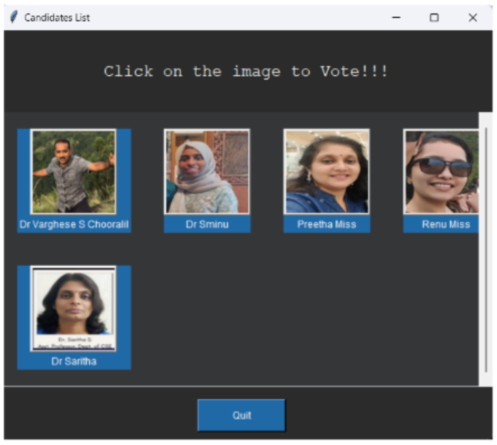
UID	NAME	IMAGE
90	Dr Varghese S Chooralil	
23	Dr Sminu	
24	Preetha Miss	
25	Renu Miss	
27	Dr Saritha	



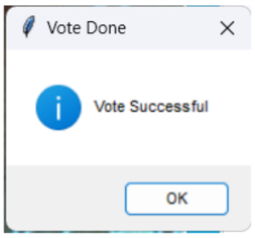
Voters List

UID	NAME	IMAGE
3	abel	
qwerfd	amal	
29	Alwin	
48	Athif	
98	Jennifer	

Voting Window

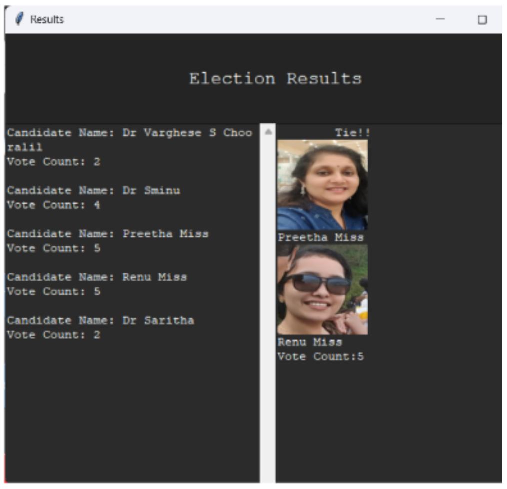


Click on the image to Vote!!!



Vote Successful

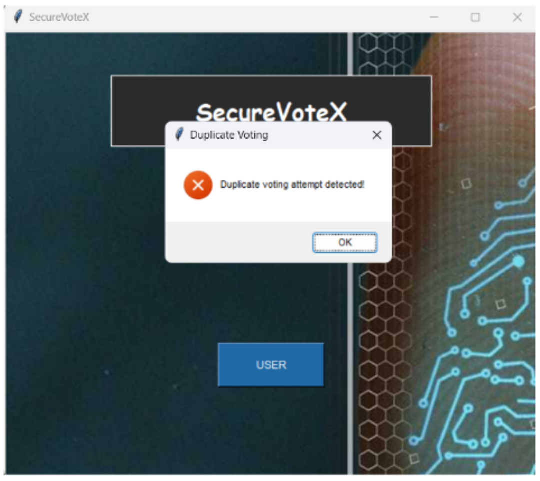
Election Result



```
Election Results

Candidate Name: Dr Varghese S Choo
ralil
Vote Count: 2

Candidate Name: Dr Sminu
Vote Count: 4

Candidate Name: Preetha Miss
Vote Count: 5

Candidate Name: Renu Miss
Vote Count: 5

Candidate Name: Dr Saritha
Vote Count: 2
```

```
Tie!!
Preetha Miss
Renu Miss
Vote Count:5
```

Duplicate Voting



Duplicate voting attempt detected!

Chapter 9

Risks and Challenges

The fingerprint-based voting system, while offering unique advantages, also faces certain risks and challenges. Some of these include:

1. False Acceptance/Rejection: There is a risk of false acceptance, where an unauthorized person's fingerprint is mistakenly identified as a legitimate voter, or false rejection, where a valid voter's fingerprint is not recognized. This can lead to compromised authentication and affect the accuracy of the voting process.

2. Biometric Accuracy: The accuracy of fingerprint recognition technology can vary based on factors such as the quality of the fingerprint sensor, environmental conditions, and variations in individuals' fingerprints. Ensuring consistently accurate identification across diverse population groups can be challenging.

3. Privacy Concerns: The collection and storage of individuals' fingerprints raise privacy concerns. Safeguarding the fingerprint data from unauthorized access or misuse is crucial to maintain public trust in the system.

4. Vulnerability to Spoofing: Fingerprint sensors can be vulnerable to spoofing techniques, such as fake fingerprints or lifted prints. Advanced spoofing techniques could potentially deceive the system, compromising the integrity of the voting process.

5. Technical Infrastructure: Implementing a fingerprint-based voting system requires a robust technical infrastructure, including reliable fingerprint sensors, secure databases, and efficient connectivity. Maintaining and upgrading this infrastructure can be costly and complex.

6. Accessibility and Inclusivity: Fingerprint-based systems may pose challenges for individuals with certain disabilities, such as those with missing or damaged fingerprints. Ensuring equal accessibility and inclusivity for all voters is essential.

7. Adoption and Trust: Introducing a new voting system requires gaining public trust and acceptance. Transparency, rigorous testing, and clear communication about the system's security measures are necessary to address concerns and build confidence.

Addressing these risks and challenges requires comprehensive security protocols, robust authentication algorithms, continuous system monitoring, and regular audits. Careful consideration and implementation of appropriate measures are crucial to ensure the integrity, accuracy, and inclusivity of a fingerprint-based voting system.

Chapter 10

Conclusion

The proposed fingerprint-based voting system aims to overcome traditional voting system problems. Its efficiency relies on the web interface's usability, ensuring a safer method for a developing nation. The system's benefits include preventing access to illegal voters, transparency, and maintaining voting integrity. It also solves the issue of rigging, as each user's fingerprint is recorded once per election. The system reduces invalid votes, polling time, and the need for extensive staff. It provides an interface for registered users to view candidates and vote securely. The fully automated system guarantees quick, error-free results, protecting against vote manipulation by admins. Additionally, it saves time and resources compared to traditional methods.

The present study utilizes one biometric module for user verification, with plans to integrate additional modules, including an Iris Scanner and face detection technology. This multi-modal approach aims to enhance voting system authenticity and security. Face detection will enable users from any part of the country to cast their votes remotely, fostering inclusivity and increasing democratic participation. The system will combine fingerprints, iris patterns, and facial features for robust authentication, preventing fraudulent activities and ensuring the integrity of the voting process. Data privacy and security measures will be prioritized to safeguard voters' biometric information. This visionary approach has the potential to revolutionize the electoral landscape and empower citizens nationwide.

References

[1] The website of Election commission of India. https://eci.gov.in/

[2] Help for Microsoft Visual Studio. https://visualstudio.microsoft.com/

[3] https://ieeexplore.ieee.org/abstract/document/7972261

[4] https://ieeexplore.ieee.org/document/8261341

[5] https://ieeexplore.ieee.org/document/6545943

[6] https://ieeexplore.ieee.org/document/5734969

[7] https://ieeexplore.ieee.org/document/7073090

[8] Electrical system-A comprehensive introduction by David M Farrell (Page 183 to page 187).

[9] https://theleaflet.in/the-urgent-need-to-modernise-indian-elections/

[10] https://indianexpress.com/article/cities/lucknow/bjp-winning-by-fraud-in-counting-not-by-votes-akhilesh-8608050/

[11] https://indianexpress.com/article/cities/lucknow/bjp-winning-by-fraud-in-counting-not-by-votes-akhilesh-8608050/

[12] R307 hardware documentation , Rajguru Electronics (I) Pvt. Ltd

Appendix A: Sample Code

main.py

```
from tkinter import *
import PIL.Image
from PIL import *
import sqlite3
from fcon import loading_window
from fcon import f
from admin import *
from user import *
#from admin1 import *
con = sqlite3.connect(database="SecureVoteX.db")
cur = con.cursor()
loading_window()
root= Tk()
root.title("SecureVoteX")
root.minsize(width=600,height=500)
root.geometry("600x500")

# Adding a background image
background_image =PIL.Image.open("finger.jpg")
[imageSizeWidth, imageSizeHeight] = background_image.size

img = ImageTk.PhotoImage(background_image)
Canvas1 = Canvas(root)
Canvas1.create_image(300,340,image = img)
Canvas1.config(bg="#11262b",width = imageSizeWidth, height = imageSizeHeight)
Canvas1.pack(expand=True,fill=BOTH)

headingFrame1 = Frame(root,bg="#FFFFFF",bd=0.5)
headingFrame1.place(relx=0.2,rely=0.1,relwidth=0.6,relheight=0.16)

headingLabel = Label(headingFrame1, text="SecureVoteX", bg='#2b2b2b', fg='white', font=('Comic Sans
MS',20,'bold'))
headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)
def adminPage():
    ## Tries to initialize the fingerprint sensor
    try:
        #f = PyFingerprint('COM7', 57600, 0xFFFFFFFF, 0x00000000)

        if not f.verifyPassword():
            raise ValueError('The given fingerprint sensor password is wrong!')

    except Exception as e:
        messagebox.showerror('Error', 'Fingerprint sensor could not be initialized!\nException message: ' +
str(e),parent=root)
        return

    ## Search for a fingerprint and calculate hash
    try:
        messagebox.showinfo('Fingerprint Authentication', 'Waiting for fingerprint...',parent=root)

        while not f.readImage():
            pass

        f.convertImage(FINGERPRINT_CHARBUFFER1) #image buffer, char buffer

        result = f.searchTemplate()
```

```
            positionNumber = result[0]
            accuracyScore = result[1]

            if positionNumber == -1:
                messagebox.showerror('Authentication Failed', 'Fingerprint authentication failed. No match 
found!',parent=root)
                return
            elif positionNumber == 1 or positionNumber == 7:
                messagebox.showinfo('WELCOM ADMIN !!!', 'Fingerprint authentication successful!\nWELCOM ADMIN ' + 
                                    str(positionNumber),parent=root )
                Admin()
                '''admin_window = None  # Global variable to track admin window instance
                def open_admin_window():
                    global admin_window

                    if admin_window is None:
                        admin_window = Admin()
                    else:
                        admin_window.focus_set()
                open_admin_window()'''
            else:
                messagebox.showerror('Authentication Failed', 'Fingerprint authentication failed. No match 
found!',parent=root)
                return
                # Continue with admin tasks


        except Exception as e:
            messagebox.showerror('Error', 'Fingerprint authentication failed!\nException message: ' + str(e),parent=root)


btn2 = Button(root,text="ADMIN",bg='#1f69a6', fg='white',command=adminPage)
btn2.place(relx=0.4,rely=0.4, relwidth=0.2,relheight=0.1)

btn5 = Button(root,text="USER",bg='#1f69a6', fg='white',command=userPage)
btn5.place(relx=0.4,rely=0.7, relwidth=0.2,relheight=0.1)

root.mainloop()
```

candidates.py

```
from PIL import Image, ImageTk
import hashlib
import sqlite3
import tkinter as tk
from tkinter import messagebox as m
existing_root = None
con = sqlite3.connect(database="SecureVoteX.db")
cur = con.cursor()
candTable = "candidates"
encryptTable = "encryptTable"
#voteTable = "votes"

def cand():
    candidate_buttons = []
    #count = 0
```

```
def encrypt_data(data):
    hashed_data = hashlib.sha256(data.encode()).hexdigest()
    return hashed_data

def btn_click(candidate_id):
    encrypted_id = encrypt_data(candidate_id)
    print("Encrypted Candidate ID:", encrypted_id)

    try:
        # Check if the encrypted ID already exists in the table
        cur.execute("SELECT * FROM " + encryptTable + " WHERE encryptid=?", [encrypted_id])
        result = cur.fetchone()

        if result:
            # If the encrypted ID exists, update the vote count
            enc_vote = result[1] + 1
            cur.execute("UPDATE " + encryptTable + " SET count = ? WHERE encryptid = ?", (enc_vote, encrypted_id))
        else:
            # If the encrypted ID doesn't exist, insert a new row
            cur.execute("INSERT INTO " + encryptTable + " VALUES (?, 1)", (encrypted_id,))

        con.commit()
        m.showinfo("Vote Done", "Vote Successful",parent=root)
        from user import userPage
        root.destroy()
        userPage()
    except Exception as e:
        m.showerror("Failed to store encrypted ID", str(e),parent=root)

def count_votes():
    try:
        cur.execute("SELECT encryptid, count FROM " + encryptTable)
        #cur.execute("SELECT encryptid, COUNT(*) as vote_count FROM " + encryptTable + " GROUP BY encryptid")
        rows = cur.fetchall()
        for row in rows:
            encryptid = row[0]
            vote_count = row[1]
            print("Encrypted ID:", encryptid)
            print("Vote Count:", vote_count)
            print("")

        #m.showinfo("Vote Counting", "Vote counting completed")

    except Exception as e:
        m.showerror("Failed to count votes", str(e),parent=root)
global existing_root  # Declare the variable as global to modify it inside the function

if existing_root and existing_root.winfo_exists():  # Check if the existing root exists and is not closed
    existing_root.destroy()
root = tk.Toplevel()
root.title("Candidates List")
root.geometry("600x500")

'''Canvas1 = tk.Canvas(root)
Canvas1.config(bg="#12a4d9")
Canvas1.pack(expand=True, fill=tk.BOTH)'''
```

```
    main_frame = tk.Frame(root, bg="#2b2b2b")
    main_frame.pack(expand=True, fill=tk.BOTH)

    white_frame = tk.Frame(main_frame, bg="#353638")
    #white_frame.pack(padx=20, pady=20, expand=True, fill=tk.BOTH)
    white_frame.place(relx=0, rely=0.2, relwidth=1, relheight=0.67)
    canvas = tk.Canvas(white_frame, bg="#353638")
    canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

    scrollbar = tk.Scrollbar(white_frame, command=canvas.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

    canvas.config(yscrollcommand=scrollbar.set)
    canvas.bind('<Configure>', lambda e: canvas.configure(scrollregion=canvas.bbox("all")))
    candidate_frame = tk.Frame(canvas, bg="#353638")
    canvas.create_window ((0, 0), window=candidate_frame, anchor='nw')
    headingFrame1 = tk.Frame(root, bg="#2b2b2b", bd=5)
    headingFrame1.place(relx=0, rely=0, relwidth=1, relheight=0.2)

    headingLabel = tk.Label(headingFrame1, text="Click on the image to Vote!!!", bg='#2b2b2b', fg='white',
font=('Courier', 15))
    headingLabel.place(relx=0, rely=0, relwidth=1, relheight=1)

    getCand = "SELECT * FROM " + candTable

    rows_in_row = 4
    row_index = 0
    column_index = 0
    try:
        cur.execute(getCand)
        con.commit()
        for i, candidate in enumerate(cur):
            candidate_id = candidate[0]  # Assuming the candidate ID is stored in the first column of the candidate table
            candidate_name = candidate[1]
            image_path = candidate[2]  # Assuming the image path is stored in the third column of the candidate table

            # Load and resize the image
            image = Image.open(image_path)
            image = image.resize((100, 100), Image.ANTIALIAS)

            # Create a PhotoImage object from the resized image
            photo = ImageTk.PhotoImage(image)

            frame = tk.Frame(candidate_frame, bg="#1f69a6")
            frame.grid(row=i // rows_in_row, column=column_index, padx=20, pady=20, sticky="nsew")

            # Create a label with the candidate's image as a button
            image_button = tk.Button(frame, image=photo,activebackground="green", command=lambda
id=candidate_id: btn_click(id))
            image_button.image = photo  # Store a reference to the PhotoImage object to avoid garbage collection
            image_button.pack()

            # Create a label with the candidate's name
            name_label = tk.Label(frame, text=candidate_name, fg='white',bg='#1f69a6')
            name_label.pack()

            # Append the button to the candidate_buttons list
            candidate_buttons.append(image_button)

            column_index += 1
            if column_index >= rows_in_row:
                column_index = 0
                row_index += 1
    except Exception as e:
        m.showerror("Failed to fetch candidates from database", str(e),parent=root)


    def count_votes_and_quit():
        root.destroy()
        count_votes()


    quitBtn = tk.Button(root, text="Quit", bg='#1f69a6', fg='white', command=count_votes_and_quit)
    quitBtn.place(relx=0.4, rely=0.9, relwidth=0.18, relheight=0.08)
    root.protocol("WM_DELETE_root", on_close)
    existing_root = root
    root.mainloop()

    count_votes()
def on_close():
    global existing_window  # Declare the variable as global to modify it inside the function
    if existing_window:
        existing_window.destroy()  # Destroy the existing window
    existing_window = None

#cand()
```

Appendix B: CO-PO And CO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	**DESCRIPTION**	**Blooms' Taxonomy Level**
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.