Major Project Report

on

Two Level Biometric Security System for Voting

Submitted in partial fulfillment of the requirement

for the award of degree of

Bachelor of Technology

in

Electronics and Communication Engineering

Guided By:

Ms. Bindu E.

Assistant Professor

ECE Department

Submitted By:

Avinash Kumar (00110407317)

Ritika Garg (02210402816)

Snigdha Srivastava (2810402816)

Yashika Kalra (40710402816)

Department of Electronics and Communication Engineering

Amity School of Engineering and Technology, Delhi

Guru Gobind Singh Indraprastha University

2016-2020

CERTIFICATE

This is to certify that the report entitled **Two Level Biometric Security System for Voting** submitted by Avinash Kumar (00110407317), Ritika Garg (02210402816), Snigdha Srivastava (02810402816) and Yashika Kalra (40710402816) have been accomplished under my guidance.

This report is submitted for partial fulfillment of award of degree of Bachelor of Technology in Electronics and Communication Engineering at Amity School of Engineering and Technology, affiliated to Guru Gobind Singh Indraprastha University, New Delhi.

Date: 21 Jul, 2020

(Bindu. E)

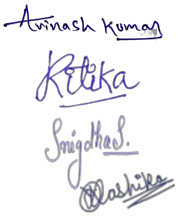
Ms. Bindu E.
Assistant Professor
E.C.E. Department
ASET, Delhi

ACKNOWLEDGEMENT

We take this opportunity to express profound gratitude and deep regards to **Prof. Dr. Rekha Aggarwal, Director, ASET-Delhi and Mr. Sanjeev Kumar, Head of Department (ECE/ICE)** who gave us this golden opportunity to learn through this research for the project.

We also take this opportunity to express a deep sense of gratitude to our guide **Ms. Bindu E.** for her cordial support, valuable information and guidance, which helped us in completing this project through various stages. The blessing and help given by her from time to time shall carry us a long way in the journey of life on which we are about to embark.

We are obliged to our project coordinators **Dr. Nidhi Sindhwani and Ms. Neetika Yadav** for the valuable information provided by them in their respective fields. We are grateful for their cooperation.



Avinash Kumar (00110407317)

Ritika Garg (02210402816)

Snigdha Srivastava (02810402816)

Yashika Kalra (40710402816)

ABSTRACT

India being the largest democracy faces a lot of issues during elections. Lot of controversies are reported about the voting system, voting machines, authentication of voting, corruptions during elections etc. Voting is one of the fundamental rights of every citizen of a democratic country. By utilizing the right of the voting, people elect their most suitable leader who will lead them. In this modern era where technology is being used in every aspect of life, election is a place to apply the best technology.

To improve the security performance in the voting process as well as to provide easy access to cast the vote by using fingerprints. Fingerprint is one of the unique identities of a human being which is being used in the Aadhar system which includes multiple layers of verifications to ensure the reliability of the device.

In the system, the authentication of the voter is done before entering the voting booth, as well as, the system does not allow a voter to cast multiple votes, thus assuring the aim of 'One voter – One vote'. The system based Aadhar like facilitated biometric possessing a Two-Tier fingerprint security. With the inclusion of a biometric fingerprint sensor, each voter is entered into the system only after being recognized and checked with the given database of enlisted voters. The voter scans his finger at the entrance gate. If the fingerprint of the voter is matched with the stored finger id in the database, the voter is an Authorized voter and the door will open. The voter will be allowed to proceed to the second level system where the voter again is asked to verify his fingerprint for image matching with the complete details stored on the system. If the details match with the authenticated voter then only he/she can cast his vote. After casting his/her vote, the voter again tries to cast a vote for the second time, he/she would be caught as the buzzer will start buzzing. Thus the system eliminates multiple voting which makes the system more secure.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

In a democratic system of governance, election is very crucial and the integrity of the electoral process is sacrosanct. Voting through an election forms an important part of democracy and for democracy to be sustainable, the voter's participation is a key consideration. Apart from voters being encouraged to exercise this democratic right, the election that facilitates the function must be credible, watertight and free of bias.

While voting, the first thing which concerns us is security. Security has become the primary concern in elections. To overcome it manually has become a real challenge for everyone. Instead, we found an alternative way which can help us to overcome our needs as well as automated. In this automated world where technology is transforming very fast one can have access to information, databases sitting at any part of the world.

A scheme for voter registration, enrolment and voting in a biometric voting system is proposed. A matching algorithm technique for facilitating the search of a matching identity to an input fingerprint is incorporated and a web based secure voting system with fingerprint authentication is implemented. A public voting system based on a biometric fingerprint method to make the election process transparent and efficient is implemented. The challenges existing in the traditional system are analyzed for using biometric authentication based Voting System.

1.1 OBJECTIVE

The objective of voting is to allow candidates to exercise their right to communicate their choices related to specific issues, pieces of law formulation, citizen initiatives, constitutional minor changes, recalls and/or to choose their government and party representatives.

Technology is being used as a tool to help voters to register their votes. The voting system is based on a two level biometric secure system . The system is able to identify each voter by getting their fingerprint. Whenever the system will receive a fingerprint, it will match the

fingerprint from the database. According to the information given by the database, the system will decide if the person is registered or not. System is also able to distinguish the second vote. If a particular voter is not a registered voter or tries to cast more than one vote, the system will identify him and will restrict him from voting as the buzzer starts buzzing. However, if neither case is applicable for a voter, it will allow the voter to cast the vote. It has a very high accuracy rate in case of both identifying voters.

To allow the exercise of this right, almost all the voting systems around the world incorporate the following steps:

- Voter identification and authentication.
- Voting and recording of votes cast.
- Vote counting.
- Publication of election results.

1.2 TRADITIONAL VOTING PROCESS

The key concerns of elections and essence of a voting system is Transparency: ordinary voters should be able to understand and observe the vote casting and counting process, even with relatively nominal education as well as Trust.

Traditional voting process consists of the following stages:

• **Voting Authentication:** This achieves our objective of ensuring a secured method of validating voters before they are allowed to cast their votes. In this phase, a voter presents his/her identity card to prove his/her identity for authentication purposes; this step is public and verified by the presiding officer. Voters would be authenticated by verifying the unique identity details possessed by them. At the end of the authentication process, the presiding officer gives a ballot paper to the voter to cast his or her vote.

• **Vote:** The vote takes place in a protected booth where voters cannot be seen by any person. The voter casts their vote by writing it with a pen on the paper ballot, folds the ballot paper and puts it into the ballot box where all the votes are mixed.

• **Vote counting:** At the end of voting time, the presiding officer deposits the ballot box containing ballots at the counting centre for counting of votes. After that with the help of members of the election committee nominated by the election commission of India, the ballot boxes are opened and votes are counted and the results are then announced.

• **Data Collection and Verification:** Various types of verification processes are used, most procedures are public and verified by the representatives of candidates of competing parties. Recount is also possible if there is any fraud or error. Before voters (staff) can be verified, there's a need for pre-recorded data in the memory i.e. database of the system. Consequently, the data of every prospective voter (staff) needs to be collected to allow verification.

This includes all necessary details needed to effectively create an election ID for them as well as enable the proposed system to determine their eligibility for various elections.

Conventional voting systems are not efficient due to long periods of preparation, bogus voting, including papers, punch cards, mechanical levers, optical-scan machines. These systems are not efficient as they are conducted manually and therefore very often are not accurate. As a consequence, it is obligatory to carry the available voting through a biometric system.

1.3 SECURE VOTING SYSTEM REQUIREMENT

In general terms, the success of any democratic system in the world depends upon the faith of the voters in the system itself. Therefore, any voting system or technology must address and adhere to the electoral requirements. In this regard, it is the responsibility of legislative or administrative bodies to set up the requirements of an electoral process. They must make sure that requirements are scientific and do not create unfairness. The fundamental qualities should be maintained. Once a suitable set of fundamental system requirements are agreed upon, the unsettling vulnerabilities of the conventional elections systems will become apparent. The importance of electoral requirements is to minimize present and future risks.

The requirements of secure voting systems are mentioned as follows:

- **Eligibility**: This seeks to accommodate all eligible voters in terms of ensuring that they all have access to a user-friendly system. Only eligible voters are allowed to cast their vote.
- **Uniqueness:** No voter is allowed to cast their vote more than once.
- **Fairness**: All eligible voters should be counted in the final tally.
- **Timeliness and accuracy**: Timeliness is based on the need to ensure that information is recorded and available results released as quickly as possible, while accuracy focuses on the ability to ensure that each individual's vote is recorded and counted.
- **Secrecy and privacy**: All participants must be permitted to vote in secret. No person can access the information about the voters vote. It must also discourage both vote-buying and coercion. Voters' privacy must also be protected in such a way that no one can discover how someone has voted except by a court order. Moreover, this will prevent electoral authorities from selling votes of the voters.
- **Security**: Usually, human beings operate computer machines. Therefore, it may be easy for them to manipulate and interfere with the whole system. In essence, it guards against manipulation and interference, using the following mechanisms: Personal identification number (PIN) or password, encryption, digital signature, biometric identifiers (like fingerprints).
- **Authentication and verification**: With regard to authentication, it should be ensured that individuals cannot be impersonated during voting. The need for verification emphasizes that the e-voting system must be able to independently verify that all voters have been correctly counted in a system.

1.4 MODE OF OPERATION FOR BIOMETRIC VOTING SYSTEM

Most biometric technology systems use the same basic principles of operation. First, a person must be registered, or enrolled, on the biometric system.

1. Enrollment: The process by which a voter's biometric data is initially acquired, accessed, processed, and stored in the form of a template for ongoing use in a biometric system is

called enrollment. Subsequent verification and identification attempts are conducted against the templates generated during enrollment.

2. Presentation: It is a process by which a user provides biometric data to an acquisition device-the hardware used to collect biometric data. Depending on the biometric system, presentation i.e voter may require placing a finger on a fingerprint sensor.

3. Biometric data: The biometric data users provide in an unprocessed image or recording of a characteristic. Biometric data provided by the user during enrollment and verification is used to generate biometric templates. Thus Biometric systems do not store biometric data-systems and use data for template creation. Enrollment requires the creation of an identifier such as a username or ID. This identifier is normally generated by the user or administrator during entry of personal data. When the user returns to verify, he or she enters the identifier, and then provides biometric data. Once biometric data has been acquired, biometric templates can be created by a process of feature extraction.

4. Feature extraction: The automated process of locating and encoding distinctive characteristics from biometric data in order to generate a template is called feature extraction. Feature extraction takes place during enrollment and verification-any time a template is created. The feature extraction process includes filtering and optimization of images and data in order to accurately locate features. For example, finger-scan technologies often thin ridges present in a fingerprint image Since quality of feature extraction directly affects a system's ability to generate templates, it is extremely important to the performance of a biometric system.

1.5 PURPOSE OF BIOMETRIC VOTING SYSTEM

One of the latest technologies is the fingerprint biometrics system which is now considered as one of the most efficient and trusted security systems for voting. The main reason for its reliability is that a fingerprint cannot have a positive match with someone else who is an unauthorized user. Our project represents a model for a secure voting system using fingerprint biometric techniques. A framework for a secure voting system based on biometric verification is presented and implemented. This ensures secured identification and

authentication processes for the voters and candidates through the use of fingerprint biometrics.

Fingerprint biometric technique is most commonly used for identification worldwide. This is largely due to its easy and cost effective integration in existing and upcoming technologies. The integration of biometric with electronic voting machines undoubtedly requires less manpower, have much time for voters and personnel, eliminate rigging, ensure accuracy, transparency and fast results in elections.

Fingerprint similarity match is one of the most popular and authentic biometric techniques used in automatic personal identification. There are two main stages during the use of fingerprints authentication: fingerprint verification and fingerprint identification.While the goal of fingerprint verification is to verify the identity of a voter, the goal of fingerprint identification is to establish the identity of a voter. In a conventional biometric recognition system, the biometric template is usually kept on a central server during acceptance. The candidate biometric template accessed by the biometric device is sent to the server where the computation and matching steps are performed.

CHAPTER- 2

SYSTEM MODEL

We present the biometric voting system's model design which will include the hardware with use of software modelling tools to structure requirements. System model requirements help us to understand structural requirements thoroughly. The new system makes use of a biometric feature (fingerprint) to authenticate voters of the system. Fingerprint recognition hardware is integrated with the voting system.

2.1 VOTERS FINGERPRINTS PROCESSING

The fingerprint surface is made up of a system of ridges and valleys that serve as friction surfaces when we are gripping the objects. The surface exhibits very rich structural information when examined as an image. The fingerprint images can be represented by both global as well as local features. The global features include the ridge orientation, ridge spacing and singular points such as core and delta.However, verification usually relies exclusively on minutiae features. Minutiae are local features marked by ridge discontinuities.

There are about 18 distinct types of minutiae features that include ridge endings, bifurcations, crossovers and islands. Among these, ridge endings and bifurcation are the commonly used features. A ridge ending occurs when the ridge flow abruptly terminates and a ridge bifurcation is marked by a fork in the ridge flow. Most matching algorithms do not even differentiate between these two types since they can easily get exchanged under 9 different pressures during acquisition. Global features do not have sufficient discriminative power on their own and are therefore used for binning or classification before the extraction of the local minutiae features.

The various stages of a typical fingerprint recognition system. The fingerprint image is acquired using off-line methods such as creating an inked impression on paper or through a live capture device consisting of an optical, capacitive, ultrasound or thermal sensor. The first stage consists of standard image processing algorithms such as noise removal and smoothening. However, it is to be noted that unlike regular images, the fingerprint image represents a system of oriented texture and has very rich structural information within the

image.

Furthermore, the definition of noise and unwanted artifacts are also specific to fingerprints. The fingerprint image enhancement algorithms are specifically designed to exploit the periodic and directional nature of the ridges. Finally, the minutiae features are extracted from the image and are subsequently used for matching.

2.1.1 Fingerprint Enrollment

Fingerprint enrollment is the initial process of collecting fingerprint data from a person (enrollee) and storing the resulting data as a fingerprint template for later comparison. The following procedure describes typical fingerprint enrollment.

Enrollment Workflow Procedure

1. Obtain the enrollee's voter identification number (Subject Identifier).
2. Capture the enrollee's fingerprint using the fingerprint sensor.
3. Extract the fingerprint feature set for the purpose of enrollment from the fingerprint sample.
4. Repeat steps 2 and 3 until you have enough fingerprint feature sets to create a fingerprint template.
5. Create a fingerprint template.
6. Associate the fingerprint template with the enrollee through the Subject Identifier, such as a user name.
7. Store the fingerprint template, along with the Subject Identifier, for later comparison.

Fingerprint templates can be stored in any type of repository that you choose, but for this project – a local database is used.

2.1.2 Fingerprint Verification

Fingerprint verification is the process of comparing the fingerprint data to the fingerprint template produced at enrollment and deciding if the two match. The following procedure describes typical fingerprint verification.

Verification Workflow Procedure:

1. Obtain the Voter Identification Number of the person to be verified.
2. Capture a fingerprint sample using the fingerprint reader.
3. Extract a fingerprint feature set for the purpose of verification from the fingerprint sample.
4. Retrieve the fingerprint template associated with the Subject Identifier from your repository.
5. Perform a one-to-one comparison between the fingerprint feature set and the fingerprint template, and make a decision of match or non-match.

2.2 ELECTORAL PROCESS

Elections make a fundamental contribution to democratic government. Elections enable voters to select leaders and to hold them accountable for their performance in office. So, the process of the voting should be free from discrepancies.

2.2.1 Voter

The voter is the candidate who wants to cast his/her vote on the voting system during election to elect their respective representatives.

- **Register Profile:** This is a major functional requirement of the system. Here the voter's details as well as fingerprint template is enrolled on the system and stored in the database.
- **Complete Profile:** The voter whose fingerprint template has not been enrolled as a result of pre-loading his data, is expected to enroll only his or her fingerprint template, in order to have a complete voter profile and obtain a voter identification number.
- **Authenticate and Validate:** This is a functional requirement of the system. It entails the capturing of the voter's identification number and fingerprint image for verification against the template already stored on the database during the registration enrolment process. It does a comparison of the fingerprint templates, detecting the

minutiae points and matching them. If successful, the voter's id is stored on the list of authenticated voters for the particular election.

- **Cast Vote:** The voter is able to exercise their franchise and choose their preferred candidate for any ballot or post in an election.
- **View Result:** The voter can view the results of all elections that have been closed and stopped.

2.2.2 Registration Officer

The registration officer is an administrator who manages the operation of the Offline (or Desktop) module of the University of Ibadan Online Voting System.

1. Manage Voters: This is a major functional requirement of the system. The registration officer is the user responsible for enrolling the various voters as well as updating or deleting their registration in the Voting System. Key functions include capturing the voters enrolling the voters fingerprint template, which is stored on the file system and the database, respectively.

2. Authenticate Voters: This is the responsibility of the registration officer to verify the eligibility of a voter by entering the voter's unique identification number and if eligible, carry out the Fingerprint template verification to determine the true identity of the voter. If a voter is properly matched, the voter records as well as the election for which the voter was authenticated for, would be saved into the list of authenticated voters.

3. Manage Eligibility: The registration officer is authorized to create, edit, and delete eligibility criteria for any election that is created on the system and for which voters would be required to be authenticated for. Eligibility criteria management is a critical functional requirement of the election.

2.3 STAGES OF VOTING

The process of voting takes in two stages. In the first stage, there will be a system installed at the entrance gate, this system is composed of Arduino and a fingerprint sensor. The second

stage in which voting occurs, after entering the polling booth the voter needs to go to the voting officer present in the booth.

Voter recognition is required during two phases of the electoral procedure:

2.3.1 Entry Side

At this stage, voters need to verify their identity by putting their finger on the fingerprint scanner, a message on the LCD will be displayed 'Scan your finger!', then their fingerprint will be matched from those which are stored in the database and if the fingerprint is verified, a message will be again displayed in the LCD 'Authorized voter', and the gate of the polling booth will open and then the voter will be allowed to go inside the polling booth.

If the fingerprint is not verified, a message will be again displayed in the LCD 'Unauthorized voter', and the gate of the polling booth will remain closed. After an individual has casted his/her vote, if he/she again tries to enter the voting room they will be blocked at the entrance gate as they have already scanned their finger.

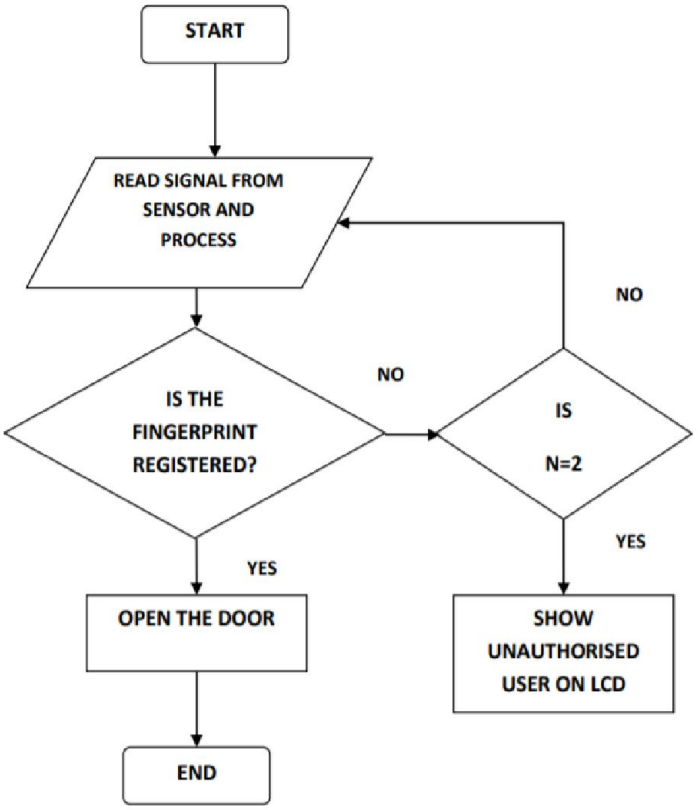


Fig 2.1 Flowchart for Entry Side

2.3.1.1 Algorithm for Entry Side

- The voter scans his finger at the entrance gate.
- The fingerprint of the voter is matched with the stored finger ID in the database.
- If the fingerprint is matched, the voter is an authorized voter, the door will open.
- The voter enters inside the door and the door will close.
- If the same finger is scanned more than once, the door doesn't open and shows error.
- If once fingerprint doesn't match with database, user gets 2 more chances, if it still doesn't match LCD shows Unauthorized Access.

2.3.2. Ballot Side

Voters will be asked to scan a finger again on the fingerprint sensor and check if the fingerprint matches with the database. If the fingerprint matches from the fingerprint which is stored in the database corresponding to the entered ID, a message on the LCD will be displayed 'VOTE NOW', then the servo motors will rotate and the gate under which voting buttons are present are made visible to the voter.

There will be options of the various parties from which a voter can choose a party of their choice by pressing the button present on the machine for their party. After voters press any one of the buttons, the LED will glow and the LCD will display " YOU HAVE VOTED FOR <NAME>" and the gate of the voting machine will automatically close. Multiple votes cannot be casted as the buzzer will start as soon as he/she presses the voting button for the second time and the culprit will be caught and the machine will not count fake votes.

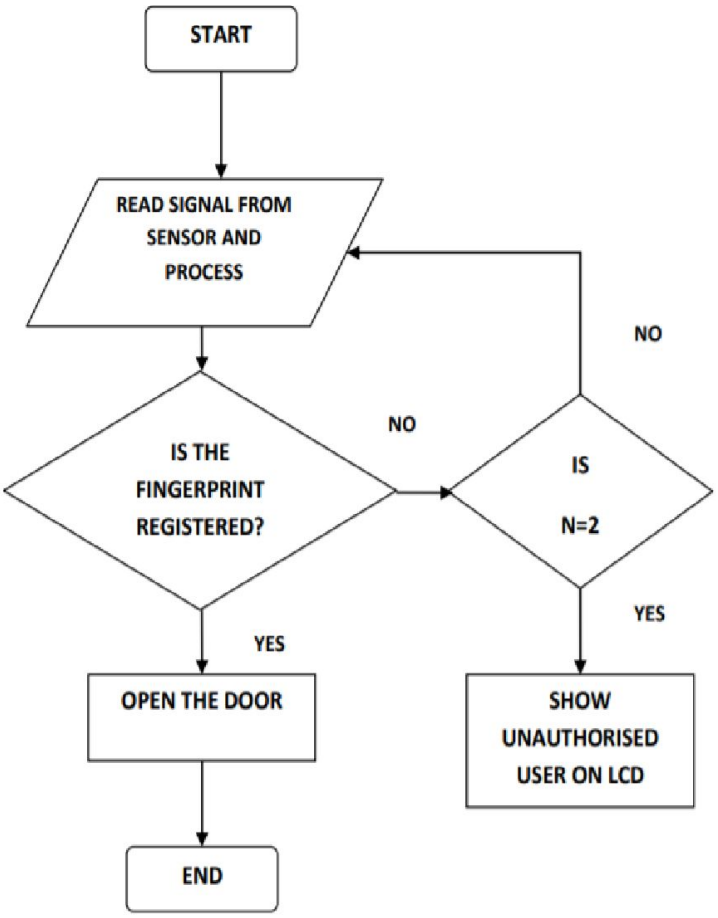


Fig 2.2 Flowchart for ballot side

2.3.2.1 Algorithm for Ballot Side

- Voters will be asked to scan a finger again on the sensor and check if the fingerprint matches with the database.
- After manually scanning the photo with the data record in our system, it will enable the voter to cast his vote.
- If the same finger is scanned more than once, the buzzer will start and the person will be caught.

CHAPTER 3
HARDWARE AND SOFTWARE

Arduino is basically the heart of this project which controls the whole project like solenoid lock, LCD and LED. Arduino provides a customary type issue that breaks the functions of the micro-controller into an additional accessible package.

3.1 HARDWARE REQUIREMENTS

1. Arduino Uno R3
2. R305 Fingerprint Module
3. Solenoid Lock
4. NPN Transistor BC557B
5. LM1084 Transistor
6. Tip 122 Transistor
7. LCD Display

3.2 ARDUINO UNO R3

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, X publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers.

For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the Processing project, which includes support for the C and C++ programming languages.

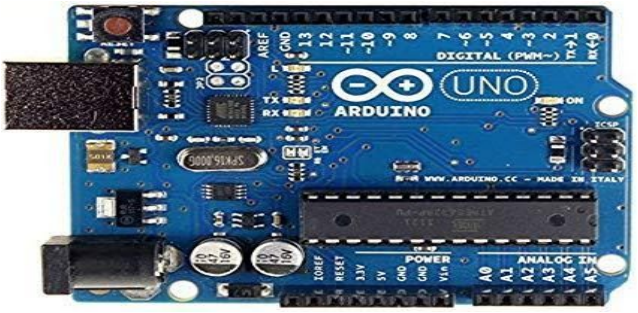


Fig 3.1 Arduino UNO

An Arduino board historically consists of an Atmel 8-,16-or 32-bit AVR microcontroller (although since 2015 other makers' microcontrollers have been used) with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as shields.

Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus - so many shields can be stacked and used in parallel. Official Arduinos use the Atmel AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280 and ATmega2560. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator.

The Arduino IDE supports the C and C++ programming languages using special rules of code organization. The Arduino IDE supplies a software library called "Wiring" from the Wiring project, which provides many common input and output procedures. A typical Arduino C/C++ sketch consist of two functions that are compiled and linked with a program stub main() into an executable cyclic executive program:

setup(): a function that runs once at the start of a program and that can initialize settings.
loop(): a function called repeatedly until the board powers off.

After compilation and linking with the GNU toolchain, also including with the IDE distribution, the Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal coding that is loaded into the Arduino board by a loader program in the board's firmware.

3.3 R305 FINGERPRINT MODULE

Fingerprint Sensor (R305) -TTL UART is a fingerprint sensor module with TTL UART interface. The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person.

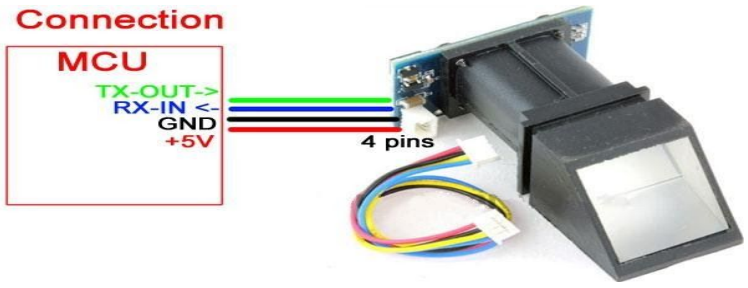


Fig 3.2 R305 Fingerprint Module

The finger print module can directly interface with a 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with a PC.

3.3.1 Features of Fingerprint Sensor (R305) -TTL UART

- Power DC : 3.6V-6.0V
- Interface : UART (TTL logic level)/ USB 1.1
- Working current : 100mA
- Peak Current : 150mA
- Matching Mode: 1:1 and 1:N
- Baud rate (9600*N)bps, N=1-12 (default N=6 57600bps)
- Character file size: 256 bytes
- Image acquiring time : <0.5s
- Template size : 512 bytes
- Storage capacity: 256
- Security level : 5 (1, 2, 3, 4, 5(highest))
- FAR : <0.001%
- FRR: <0.1%
- Average searching time: < 0.8s (1:880)
- Window dimension : 18mm*22mm

Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching (the matching can be 1:1 or 1:N). When enrolling, the user needs to enter the finger two times.

The system will process the two time finger images, generate a template of the finger based on processing results and store the template. When matching, the user enters the finger through an optical sensor and the system will generate a template of the finger and compare it with templates of the finger library. For 1:1 matching, the system will compare the live finger with a specific template designated in the Module; for 1:N matching, or searching, the system will search the whole finger library for the matching finger. In both circumstances, the system will return the matching result, success or failure.

3.4 SOLENOID LOCK

Solenoid is the generic term for a coil of wire used as an electromagnet. It also refers to any device that converts electrical energy to mechanical energy using a solenoid. The device creates a magnetic field from electric current and uses the magnetic field to create linear motion. Common applications of solenoids are to power a switch, like the starter in an automobile, or a valve, such as in a sprinkler system.

3.4.1 Working of Solenoid

A solenoid is a coil of wire in a corkscrew shape wrapped around a piston, often made of iron. As in all electromagnets, a magnetic field is created when an electric current passes through the wire. Electromagnets have an advantage over permanent magnets in that they can be switched on and off by the application or removal of the electric current, which is what makes them useful as switches and valves and allows them to be entirely automated.

Fig 3.3 Solenoid Lock

3.5 NPN TRANSISTOR-BC557B

- Collector-Emitter Voltage: **-45** V
- Type - **NPN**
- Collector-Base Voltage: **-50** V
- Emitter-Base Voltage: **-5** V
- Collector Current: **-0.1** A
- Collector Dissipation - **0.5** W
- DC Current Gain (h_{fe}) - **200** to **450**
- Transition Frequency - **150** MHz
- Noise Figure - **2** dB
- Operating and Storage Junction Temperature Range **-65 to +150** °C
- Package - **TO**

3.6 LM1084 TRANSISTOR

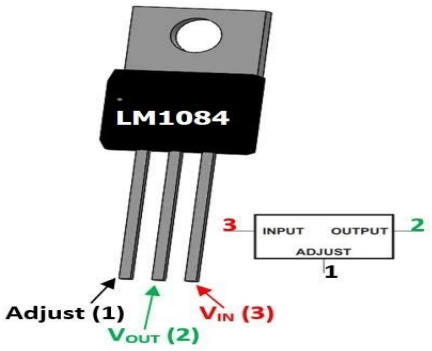


Fig 3.4 LM1084 Low Dropout Positive Regulator Pinout

Pin No.	Pin Name	Description
1.	ADJ/GND	Adjustable pin for adjusting output version. Ground for fixed type version
2.	OUTPUT	Output Voltage Pin
3	INPUT	Input Voltage Pin

3.6.1 Features

- Three variants available: 3.3V, 5V and Adjustable Version
- Current Limiting and Thermal Protection
- 5Amps of Output Current
- Line Regulation 0.015% (typical)
- Load Regulation 0.1% (Typical)
- Available Package: TO-263 and TO-220

3.7 470UF 25V CAPACITOR

Fig 3.5 470uF 25V Capacitor

- Jameco Part no.: **2280023**
- Manufacturer: **Nippon Chemi Con**
- Manufacturer p/n: **ELXV250ETD471MJ20**

3.8 LCD DISPLAY

Liquid crystal display (LCD) has material which combines the properties of both liquid and crystals. They have a temperature range within which the molecules are almost as Fingerprint Based Security System mobile as they would be in a liquid, but are grouped together in an order form similar to a crystal.

Fig 3.6 LCD Display

More microcontroller devices are using 'smart LCD' displays to output visual information. The

following discussion covers the connection of a Hitachi LCD display to a PIC microcontroller. LCD displays designed around Hitachi's LCD HD44780 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 8 x 80 pixels of the display. Hitachi LCD displays have a standard ASCII set of characters plus Japanese, Greek and mathematical symbols.

3.9 SOFTWARE REQUIREMENTS

The following software requirements of the voting system application are as follows:

- Language – C++, SQL, PHP
- Database – SQL Server
- Libraries – Fingerprint Reader SDK, Business Entities SDK
- Integration Platform – Thingspeak, Wampserver 2.4E, Netbeans 8.0.1 IDE

3.10 LANGUAGES USED

C++ is a high-level programming language developed by Bjarne Stroustrup at Bell Labs. It adds object-oriented features to its predecessor, C. C++ is one of the most popular programming languages for graphical applications. C++ is one of the most popular languages primarily utilized with system/application software, drivers, client-server applications and embedded firmware. The C++ is a collection of predefined classes, which are data types that can be instantiated multiple times. The language also facilitates declaration of user-defined classes. Classes can further accommodate member functions to implement specific functionality.

SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables. All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language. SQL allows users to query the database in a number of ways, using English-like statements. MySQL has been proven with a high performance rating as well as optimal query operations.

PHP is an HTML-embedded, server-side scripting language designed for web development. PHP codes are simply mixed with HTML codes and can be used in combination with various web frameworks. Its scripts are executed on the server. PHP code is processed by a PHP interpreter. The main goal of PHP is to allow web developers to create dynamically generated pages quickly. A PHP file consists of texts, HTML tags and scripts with a file extension of .php, .php3, or .phtml. You can create a login page, design a form, create forums, dynamic and static websites and many more with PHP. PHP was used to implement the online sub-system of this application. PHP as a language is now object – oriented thus it can serve the purpose of developing this application very well. Also, it is cheaper to deploy online, with a vast array of technical support as well as the open source nature of the language.

JAVA is an object-oriented, class-based, concurrent, secure and general purpose computer-programming language. It is a widely used robust technology. Java is a programming language and a platform**.** Java is a high level, robust, object-oriented and secure programming language. Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform. JAVA was used to implement the desktop (Offline) sub-system of this application. JAVA has a wealth of data structures and methods in its development kit or library such as the swing package which the graphical user interface of this application leverages on; also, the language is highly portable which means that it can run on any hardware architecture and it is robust as well as being highly secured.

3.11 DATABASE DESIGN

It refers to the organization of data and essentially captures the entities after normalization and includes all their properties as well as the constraints that exist on the relationships between the entities. A RDBMS, specifically MySQL Server, is used to implement the database design formulated.

3.12 LIBRARIES USED

Fingerprint Reader SDK : Fingerprint Software Developer Kit is a necessary tool for software developers to develop fingerprint recognition based applications. Fingerprint SDK

allows your application to directly interact with USB port Fingerprint scanners and execute functions through a high level Application Programming Interface (API) provided by SDK.

BioEnable SDK is a Software Development kit, combining the **e**xisting BSP (Biometric Solution Provider)SDK and 1:N fingerprint recognition engine**,** not only for a regular fingerprint application, but also for an application using fingerprint DBs of large capacity with high fingerprint search speed. It provides a High Level API (Application Programming Interface) which allows easy and swift implementation of software and UI (User Interface) of the wizard type, so that it saves the user's time and effort in developing an application. It operates on a variety of platforms supporting different operating systems and development languages.

BioEnable SDK Fast Fingerprint recognition SDK Development language support VB, C#.Net BioEnable SDK also connects to other fingerprint recognition devices, which are manufactured or based on advanced hardware design technology. It supports a single finger as well as multi finger scanners**.** It not only offers optimal conditions for development of fingerprint recognition-related solutions with a sophisticated fingerprint recognition algorithm inside, but also guarantees high fingerprint search speed and high recognition ratio.

Fingerprint SDKs provide a low-level framework to communicate with the fingerprint reader, capture an image, extract the unique minutiae data from the image, and compare two sets of extracted minutiae data. However, the developer must overcome a significant learning curve in order to produce a robust, turn-key fingerprint system from even the best fingerprint sdk.

Fingerprint SDK Functions

- **initialize()** - Initializes the fingerprint SDK API (when starting).
- **cleanup()** - Release resources initialized by API (when finished).
- **openDevice()** - Open a fingerprint scanner device.
- **closeDevice()** - Close fingerprint scanner device.
- **captureImage()** - Captures a fingerprint image from the device.
- **getMinutiae()** - Extracts minutiae points from captured fingerprints.
- **matchMinutiae()** - Compares two minutiae records (1-to-1).

3.13 INTEGRATION PLATFORM

- **Wampserver 2.4E :** WAMP is an acronym that stands for Windows, Apache, MySQL

and PHP.It's a software stack which means installing WAMP installs Apache, MySQL, and PHP on your operating system (Windows in the case of WAMP). WAMP acts like a virtual server on your computer. It allows you to test all WordPress features without any consequences since it's localized on your machine and is not connected to the web.

"**W**" stands for Windows, there's also LAMP (for Linux) and MAMP (for Mac). "**A**" stands for Apache. Apache is the server software that is responsible for serving web pages. When you request a page to be seen by you, Apache grants your request over HTTP and shows you the site. "**M**" stands for MySQL. MySQL's job is to be the database management system for your server. It stores all of the relevant information like your site's content, user profiles, etc. "**P**" stands for PHP. It's the programming language that was used to write WordPress. It acts like glue for this whole software stack. PHP is running in conjunction with Apache and communicating with MySQL.

- **Netbeans 8.0.1 IDE :** NetBeans IDE is a free, open source, integrated development

environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++.

The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the GlassFish Server, the reference implementation of Java EE.

CHAPTER 4
MATCHING ALGORITHM

Biometric authentication is a system that is used for unique physiological and behavioral characteristics of individuals for identification of a person to secure access. Biometric is not used for personal identification to establish absolute “yes” or “no” but used for identification to achieve “positive”. As uniqueness and permanence of fingerprint recognition is higher than other biometric recognitions, so we mostly use fingerprint authentication systems as a means of identification of a person. Fingerprints of a person can never be changed throughout a person's death except deep physical injuries and severe burns.

4.1 FINGERPRINT RECOGNITION

The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These include patterns, which are aggregate characteristics of ridges, and minutiae points, which are unique features found within the patterns. It is also necessary to know the structure and properties of human skin in order to successfully employ some of the imaging technologies.

4.1.1 Patterns

The three basic patterns of fingerprint ridges are the arch, loop, and whorl:

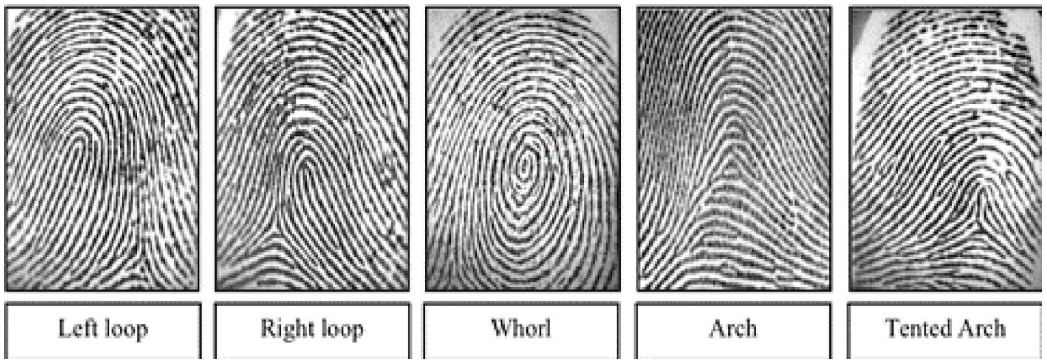


Fig 4.1 One fingerprint from each of the five major classes

Arch: The ridges enter from one side of the finger, rise in the center forming an arc, and then exit the other side of the finger.

Loop: The ridges enter from one side of a finger, form a curve, and then exit on that same side.

Whorl: Ridges form circularly around a central point on the finger.

Scientists have found that family members often share the same general fingerprint patterns, leading to the belief that these patterns are inherited.

4.1.2 Fingerprint Processing

Fingerprint processing has three primary functions: Enrolment, searching and verification. Among these functions, enrolment which captures fingerprint image from the sensor plays an important role. A reason is that the way people put their fingerprints on a mirror to scan can affect the result in the searching and verifying process. Regarding the verification function, there are several techniques to match fingerprints such as correlation-based matching, minutiae-based matching, ridge feature-based matching and minutiae-based algorithm. However, the most popular algorithm was minutiae based matching algorithm due to its efficiency and accuracy.

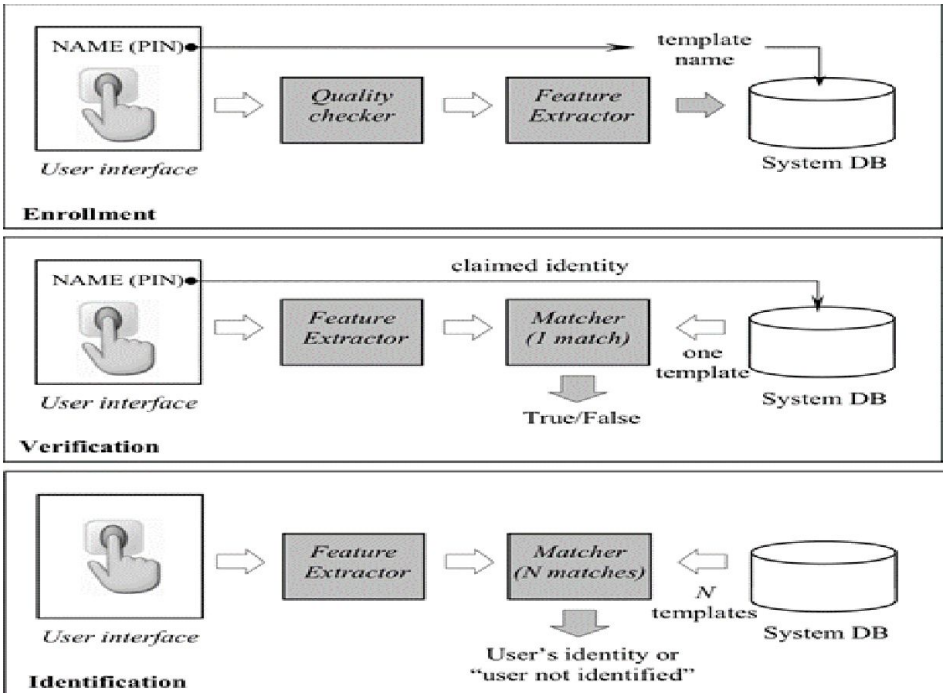


Fig 4.2 Primary functions

4.1.3 Minutiae features

The major minutiae features of fingerprint ridges are ridge ending, bifurcation, and short ridge (or dot). The ridge ending is the point at which a ridge terminates. Bifurcations are points at which a single ridge splits into two ridges. Short ridges (or dots) are ridges which are significantly shorter than the average ridge length on the fingerprint. Minutiae and patterns are very important in the analysis of fingerprints since no two fingers have been shown to be identical.

4.2 ALGORITHMS USED TO MATCH FINGERPRINT

Matching algorithms are used to compare previously stored templates of fingerprints against candidate fingerprints for authentication purposes. In order to do this either the original image must be directly compared with the candidate image or certain features must be compared.

4.2.1 Pre-processing

Pre-processing helped enhance the quality of an image by filtering and removing unnecessary noises. The minutiae based algorithm only worked effectively in 8-bit grayscale fingerprint images. A reason was that an 8-bit gray fingerprint image was a fundamental base to convert the image to 1-bit image with value 0 for ridges and value 1 for furrows. As a result, the ridges were highlighted with black color while the furrows were highlighted with white color.

This process partly removed some noises in an image and helped enhance the edge detection. Furthermore, there are two more steps to improve the best quality for the input image: minutiae extraction and false minutiae removal.

The minutiae extraction was carried out by applying a ridge thinning algorithm which was to remove redundant pixels of ridges. As a result, the thinned ridges of the fingerprint image are marked with a unique ID so that further operation can be conducted. After the minutiae extraction step, the false minutiae removal was also necessary. The lack of the amount of ink and the cross link among the ridges could cause false minutiae that led to inaccuracy in the fingerprint recognition process.

4.2.2 Pattern-based (or Image-based) Algorithm

Pattern based algorithms compare the basic fingerprint patterns (arch, whorl, and loop) between a previously stored template and a candidate fingerprint. This requires that the images can be aligned in the same orientation. To do this, the algorithm finds a central point in the fingerprint image and centers on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match.

4.2.3 Direct Matching

In this matching, the input and template images are read, the matching is performed by comparing the two images pixel wise.

4.2.4 Minutiae Extraction And Matching Algorithm

The existing fingerprint recognition systems use approaches based on the local and global feature representations of the fingerprint images such as minutiae, ridge shape, texture information, etc. The basic method of minutiae extraction is divided into three parts: Pre-processing, Minutiae Extraction, Post processing.

Step 1: Input:
In this step we take fingerprints of voter input and process them during the electoral process.

Step 2: Binarization:
Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. This transforms the 8-bit Gray fingerprint image to a 1-bit image with 0- value for ridges and 1-value for furrows.Binarization is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae.

One useful property of the Gabor filter is that it has a DC component of zero, which means the resulting filtered image has a mean pixel value of zero. Hence, straightforward

binarization of the image can be performed using a global threshold of zero. The binarization process involves examining the grey-level value of each pixel in the enhanced image, and, if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero. The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

Step 3: Thinning:

Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm is employed, which performs the thinning operation using two sub iterations. This algorithm is accessible in MATLAB via the `thin' operation under the bwmorph function. Each sub iteration begins by examining the neighbourhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub iterations continue until no more pixels can be deleted.

The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae.

Ridge thinning is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide. Uses an iterative, parallel thinning algorithm.

1) To get a thinned image we find the location of the middle black pixel at each stage of continuation of the curve.

2) In each scan of the full fingerprint image, the algorithm marks down redundant pixels in each small image window (3x3).

3) And finally removes all those marked pixels after several scans.

Step 4: Minutiae Connect:

This operation takes a thinned image as input and produces refined skeleton images by converting small straight lines to curve to maximum possible extent.

Step 5: Minutiae Margin:

This increases the margin of endpoints by one pixel of curves of length at least three pixels.

Step 6: Minutiae point Extraction:

For extracting minutiae point we compute the number of one- value of every 3x3 window:

If the centroid is 1 and has only 1 one valued neighbor, then the central pixel is a termination.
If the central is 1 and has 3 one-value neighbours, then the central pixel is a bifurcation.
If the central is 1 and has 2 one-value neighbours, then the central pixel is a usual pixel.

4.2.5 Minutiae Based Matching

This is the most popular and widely used technique, being the basis of the fingerprint comparison made by fingerprint examiners. Minutiae are extracted from the two fingerprints and stored as sets of points in the two-dimensional plane.

Fig 4.3 Minutiae Matching

Most common minutiae matching algorithms consider each minutia as a triplet m = {x, y, θ} that indicates the x, y minutia location coordinates and the minutia angle θ:

$$T = \{ m_1 , m_2, ..., m_m \}; \qquad mi = \{ x_i, y_i, \theta_i \}; \quad i = 1...m$$

$$I = \{ m_1', m_2', ..., m_n' \}; \qquad mj' = \{ x_j', y_j', \theta_j' \}; \; j = 1...n$$

where m and n denote the number of minutiae in T and I, respectively. A minutia m′j in I and a minutia mi in T are considered "matching," if the spatial distance (sd) between them is smaller than a given tolerance r0 and the direction difference (dd) between them is smaller than an angular tolerance θ0:

$$sd(m_j', m_i) = \sqrt{(x_j' - x_i)^2 + (y_j' - y_i)^2} \leq r_0 ;$$

$$dd(m_j', m_i) = min(\left|\theta_j' - \theta_i\right|, 360^0 - \left|\theta_j' - \theta_i\right| \leq \theta_0$$

The tolerance boxes (or hyper-spheres) defined by r0 and θ0 are necessary to compensate for the unavoidable errors made by feature extraction algorithms and to account for the small plastic distortions that cause the minutiae positions to change.

Aligning the two fingerprints is a mandatory step in order to maximise the number of matching minutiae. Correctly aligning two fingerprints certainly requires dis- placement (in x and y) and rotation (θ) to be recovered and likely involves other geometrical transformations like scale and specific distortion-tolerant geometrical transformations. Let map(.) be the function that maps a minutia m′j (from I) into m′ according to a given geometrical transformation; for example, by considering a j displacement of [Δx, Δy] and a counter clockwise rotation θ around the origin:

$$map_{\Delta x,\,\Delta y,\,\theta}\left(m'_j = \left\{x'_j,\, y'_j,\, \theta'_j\right\}\right) = m_j^n = \left\{x_j^n,\, y_j^n,\, \theta_j^r + \theta\right\};\ where$$

$$\begin{bmatrix} x_f^n \\ y_f^n \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x'_f \\ y'_f \end{bmatrix} + \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix}$$

Let mm (.) be an indicator function that returns 1 in the case where the minutiae m″ and mi match according to the previous equations:

$$mm\left(m_j^n,\, m_i\right) = \begin{cases} 1 & sd\left(m_j^n,\, m_i\right) \le r_0 \ and \ dd\left(m_j^n,\, m_i\right) \le \theta_0 \\ 0 & otherwise \end{cases}$$

Then, the matching problem can be formulated as:

$$maximize\ (\Delta x,\, \Delta y,\, \theta,\, P) \sum_{i=1}^{m} mm\left(map_{\Delta x,\,\Delta y,\,\theta}\left(m'_{P(i)}\right),\, m_i\right)$$

where P(i) is an unknown function that determines the pairing between I and T minutiae; in particular, each minutia has either exactly one mate in the other fingerprint or has no mate at all:

1. P(i) = j indicates that the mate of the mi in T is the minutia m′j in I.
2. P(i) = null indicates that minutia mi in T has no mate in I.
3. A minutia m′j in I, such that ∀i=1..m, P(i)≠j has no mate in T.
4. ∀ i=1..m, k=1..m, i≠k ⇒ P(i)≠P(k) or P(i)=P(k)=null (this requires that each minutia in I is associated with a maximum of one minutia in T).

CHAPTER - 5
PROJECT IMPLEMENTATION AND RESULTS

This project focuses on fingerprint recognition biometric systems based on real time embedded systems which will provide complete security solutions, therefore making it difficult for unauthorized people to access it. In comparison to other methods, authentication through RFID, and password security. This method has proven to be most efficient and reliable. This project highlights the development of a fingerprint verification system using Arduino Uno R3.

Verification is completed by comparing the data of authorized fingerprint image with incoming fingerprint image. The incoming fingerprint image will first go through the extraction and filtering processes through which the information about it is obtained. Then the information of the incoming fingerprint image will undergo the comparison process to compare it with the authorized fingerprint image. Only if the fingerprint image matches any of the templates in the database, the door lock will open.

5.1 PROJECT OVERVIEW

This project is designed to implement a system that works with 2 step authentication for the election process. With trust, credibility and voter apathy challenges constantly cropping up in the traditional manual way of elections, the development of this online voting system with biometric authentication – specifically the use of the voter fingerprints – will be a welcome innovation to the staff voting electorate. It also helps in restoring belief and encouraging more staff members to always participate and have their say in how they are governed administratively by fully exercising their voting rights or franchise.

This project delivers an efficient and secure Online Voting System with Biometric Authentication for elections to reduce, if not eradicate, the problem of eligibility, wrong voter information as well as centralized and stressful voting location.

5.2 SOFTWARE IMPLEMENTATION

The software program is written in C, compiled and uploaded using Arduino IDE 1.8.9. After compiler operation the .ino file is created and stored in the computer. On uploading, the .ino type file is saved onto the Arduino Uno board that we have used easily by defining the ports we are using for the purpose.

5.3 CODE DESCRIPTION

Below piece of code is used to take Finger Print as input and take action according to validation of finger. If the finger will be validated, the gate will be open, otherwise remain closed.

5.3.1 Setup and Loop

The Setup function: This function is used to initiate serial communication between the Arduino and the R305. The baud rate is set at 9600 for data transmission between the Arduino and the fingerprint module. The data rate for the sensor serial port is set at 57600.

```
void setup() {
        while (!Serial);
                delay(500);
        Serial.begin(9600);
        Serial.println("Adafruit Fingerprint sensor enrollment");
        finger.begin(57600);
}
```

The loop function: The loop function runs indefinitely as long as the Arduino is connected to the power supply. It displays instructions for the user to enroll his or her fingerprint in the fingerprint module and asks for an ID as input for saving the current fingerprint image.

```
void loop() {
```

```
        Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want
        on save this finger as...");
        id=readnumber();
        Serial.print("Enroling");
        while (! getFingerprintEnroll() );
}
```

5.3.2 Stages involved in Enrolment Phase

Obtaining the fingerprint image:

```
uint8_t A_Fingerprint::getImage(void) {
        uint8_t packet[] = {FINGERPRINT_GETIMAGE writePacket(theAddress,
        FINGERPRINT_COMMANDPACKET, 3, packet);
        uint8_t len = getReply(packet);
        if ((len != 1) && (packet[0] != FINGERPRINT_ACKPACKET))
                return -1;
        return packet[1];
}
```

Creating a template from the image:

```
uint8_t A_Fingerprint::createModel(void) {
        uint8_t packet[] = {FINGERPRINT_REGMODEL};
        writePacket(theAddress, FINGERPRINT_COMMANDPACKET,
        sizeof(packet)+2, packet);
        uint8_t len = getReply(packet);
        if ((len != 1) && (packet[0] != FINGERPRINT_ACKPACKET))
                return -1;
        return packet[1];
}
```

Storing the template in flash memory:

```
uint8_t A_FingerprinT::storeModel(uint16_t id) {
        uint8_t packet[] = {FINGERPRINT_STORE, 0x01, id >> 8, id & 0xFF};
        writePacket(theAddress, FINGERPRINT_COMMANDPACKET,
        sizeof(packet)+2,packet);
        uint8_t len = getReply(packet);
        if ((len != 1) && (packet[0] != FINGERPRINT_ACKPACKET))
                return -1;
        return packet[1];
}
```

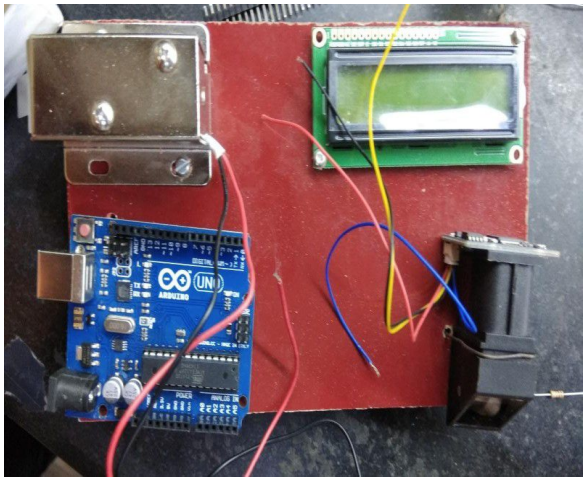
Reading a fingerprint template from flash to CharBuffer 1:

```
uint8_t A_Fingerprint::loadModel(uint16_t id) {
        uint8_t packet[] = {FINGERPRINT_LOAD, 0x01, id >> 8, id & 0xFF};
        writePacket(theAddress, FINGERPRINT_COMMANDPACKET,
        sizeof(packet)+2, packet);
        uint8_t len = getReply(packet);
        if ((len != 1) && (packet[0] != FINGERPRINT_ACKPACKET))
                return -1;
        return packet[1];
}
```

Transferring the template from CharBuffer 1 to the host computer:

```
uint8_t A_Fingerprint::getModel(void) {
        uint8_t packet[] = {FINGERPRINT_UPLOAD, 0x01};
        writePacket(theAddress, FINGERPRINT_COMMANDPACKET,
        sizeof(packet) + 2, packet);
        uint8_t len =getReply(packet);
        if ((len != 1) && (packet[0] != FINGERPRINT_ACKPACKET))
                return -1;
        return packet[1];
}
```

5.4 HARDWARE RESULTS



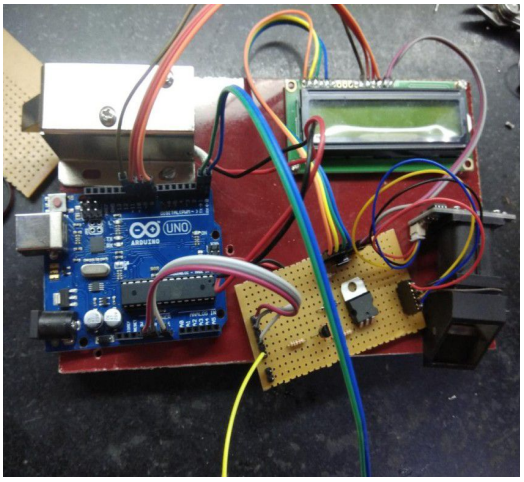


Fig 5.1 Hardware Components attached to entry side **Fig 5.2 Final connections**

5.5 SOFTWARE RESULTS

Enrolling an ID with fingerprint and matching it to unlock the solenoid in Arduino, here is the simulation process displayed in the serial monitor is given below:

STEP 1 Request for Enrolling into a database with specific ID.

STEP 2 Enroll into database done.

STEP 3 Wait for a valid fingerprint.

STEP 4 Fingerprint matched.

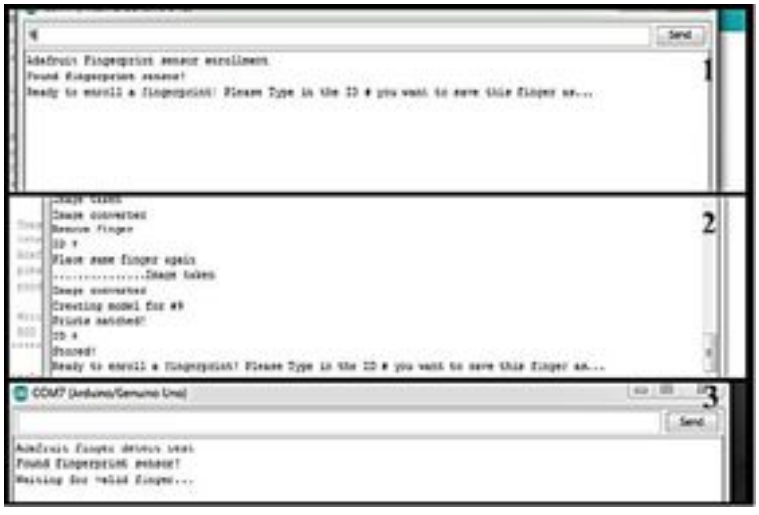


Fig 5.3 Enrollment of Fingerprint

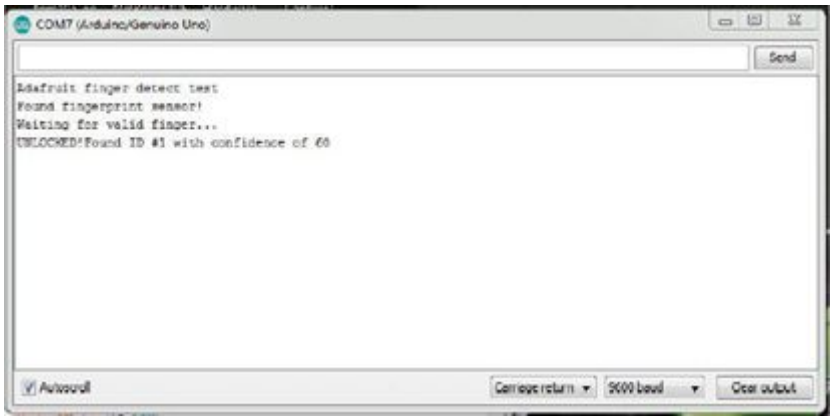


Fig 5.4 Matched Fingerprint

For easy access of all information, showing results in real time, and voting purpose, the Online Voting System website is one goto place. Following are the screenshots of all the webpages present for varied purposes.

The landing page for the website which is the Home Page welcomes the user (admin/voter) for using the Online Voting System and helps him navigate to other sections of the website namely About, Login, Complaint.

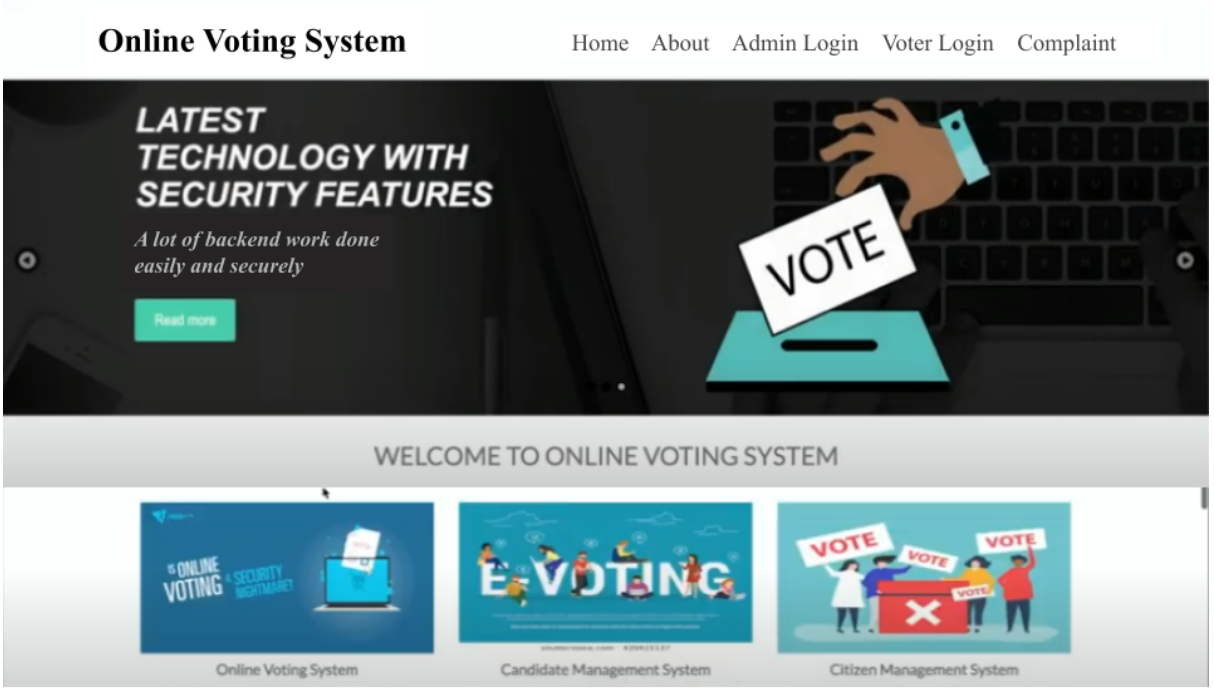


Fig 5.5 Home Page for Online Voting System

The about section briefly describes the motivation behind the website, working, and its

usefulness to the users/clients.

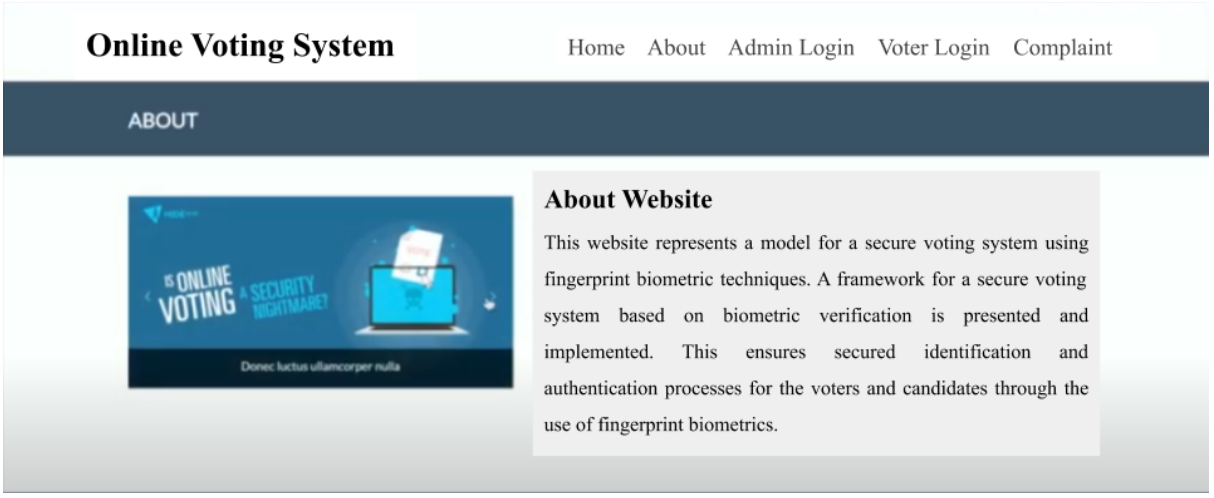


Fig 5.6 About Section of Online Voting System

The login page is the same for both the admins and the voters, but both lands at different websites with completely different displays of data and liberty of options for the users to navigate.

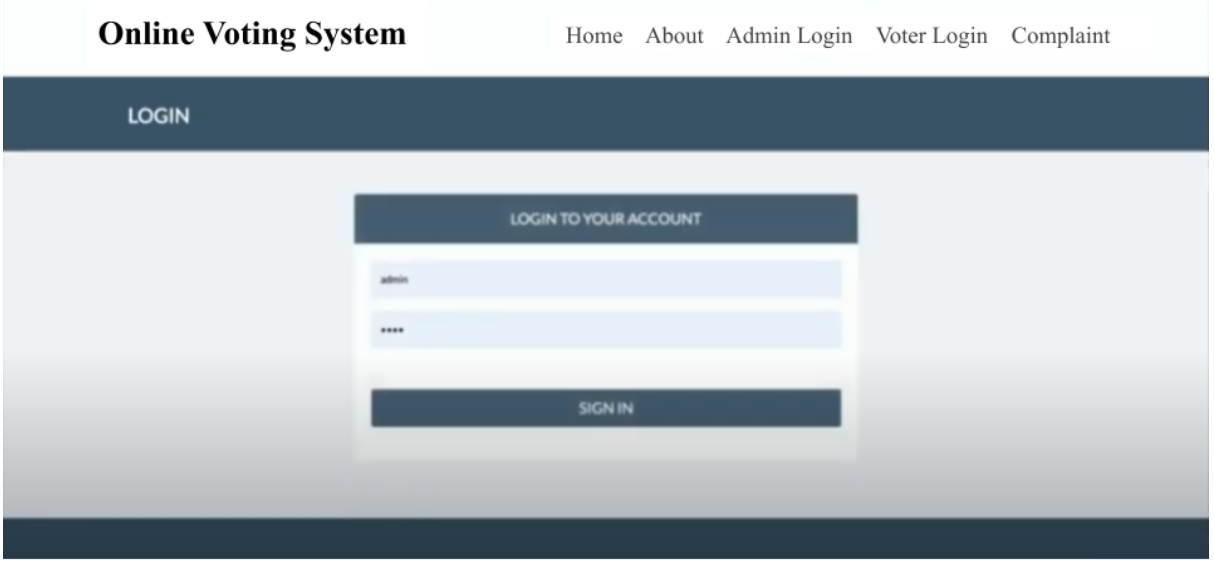


Fig 5.7 Login Page for Admin and Voter

The My Account Dropdown Section in the navigation menu on the top of the webpage takes the admins to the Election Report which is for listing all the names of people/parties sitting for elections. This section also gives admins full responsibility of adding a new name, editing existing details of a person/party or delete an existing party with one click.

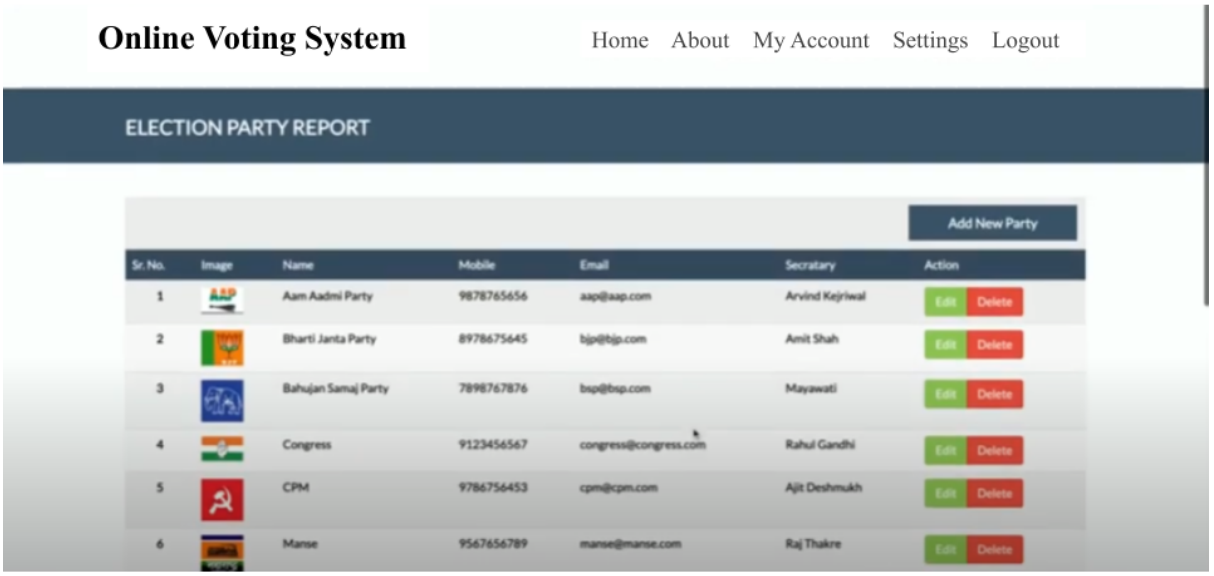


Fig 5.8 Dummy Report Page for Admins

The Election Report section has a adding a new name functionality for admins which opens up a form for the admin to fill up all the required details for the person/party to sit for the elections, checks it upon the criteria on the backend and updates the list both in the database and the frontend of the website.

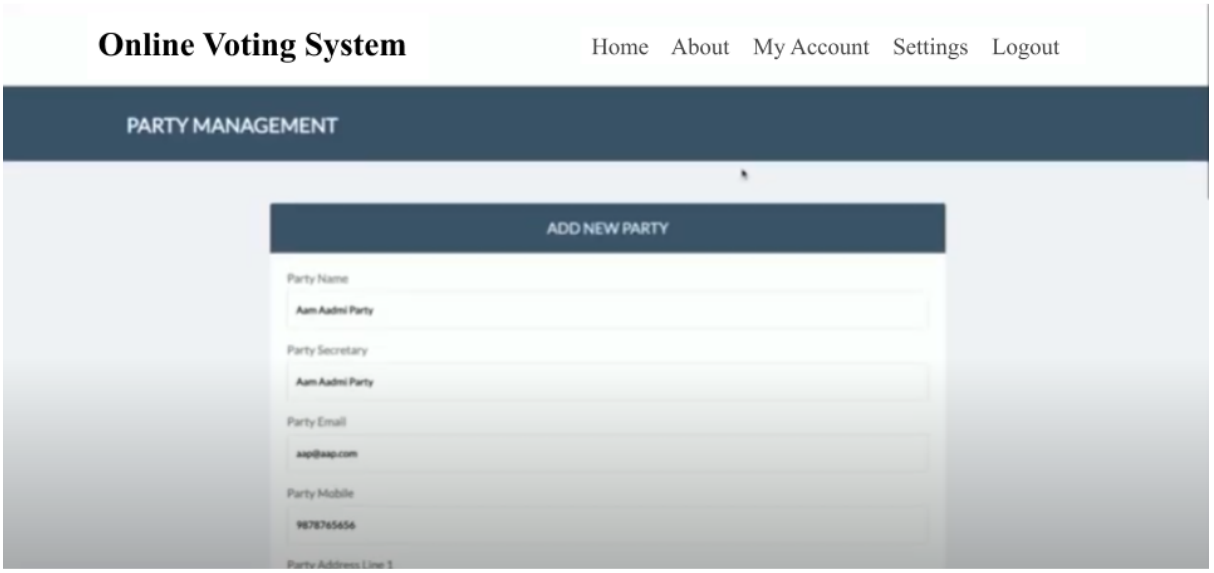


Fig 5.9 Adding New Party Form for Admins

During and after the election, to keep track of the number of votes and make sure the votes are not rigged the admins can see real time reflection of number of votes for each candidate in the Voting Result Website which can be accessed by the admins from the My Account

dropdown navigation menu.

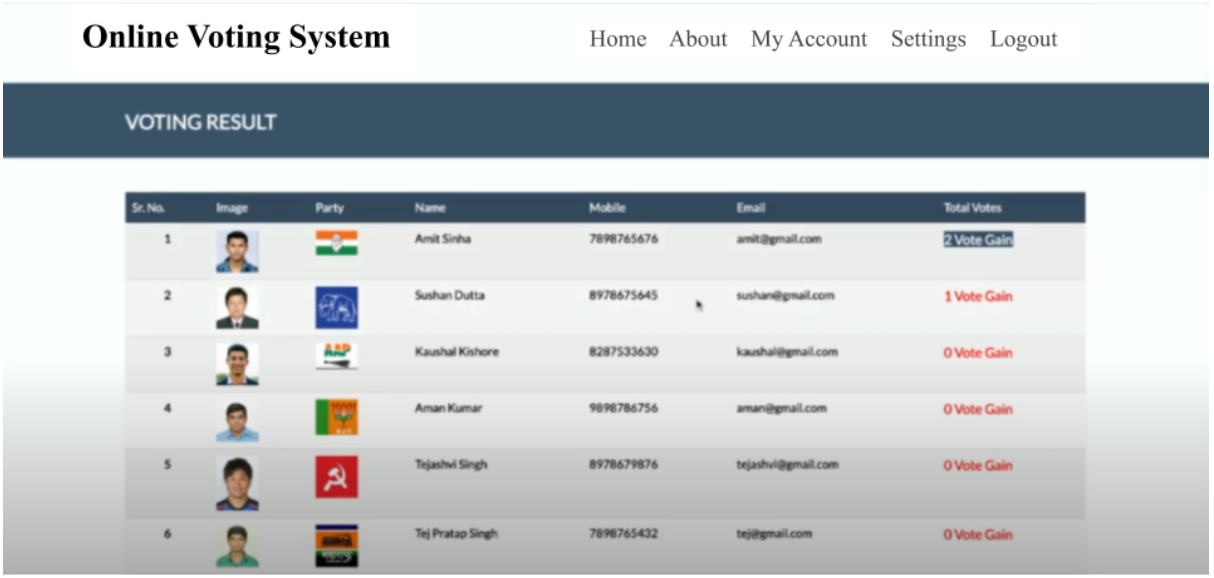


Fig 5.10 Statistics Page of the Voting Result

The user can also change their password by filling up a short form and giving the new password to the database.

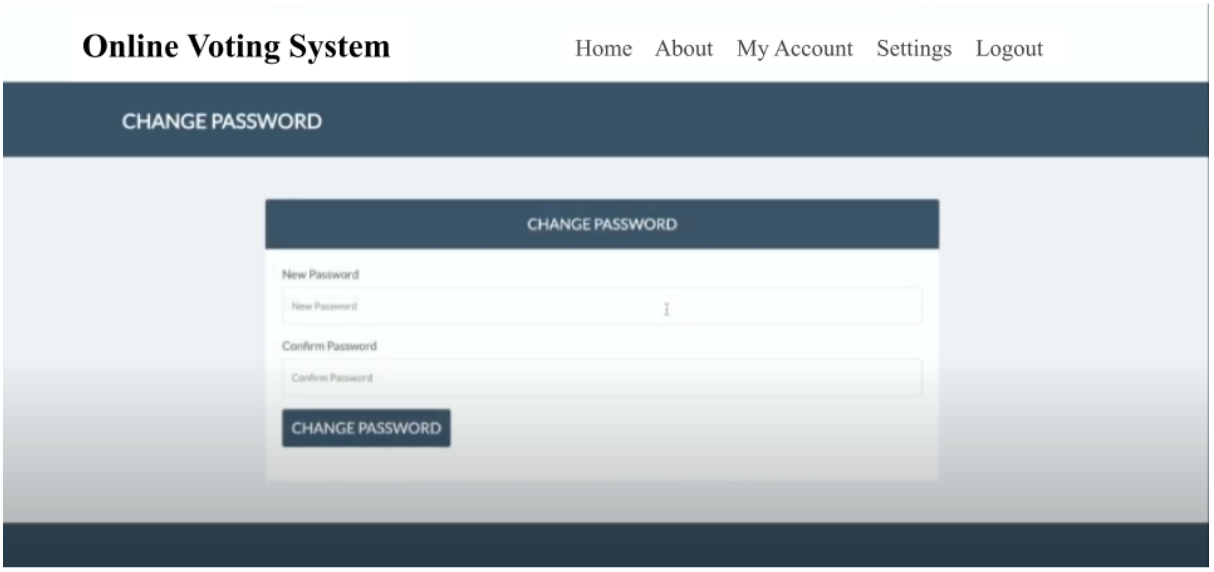


Fig 5.11 Change Password Page for Admins/Voters

The Complaint Sections gives the user details like Address, Business hours, and telephone for reach. It also gives users the liberty to drop a message with the internal form present on the website.

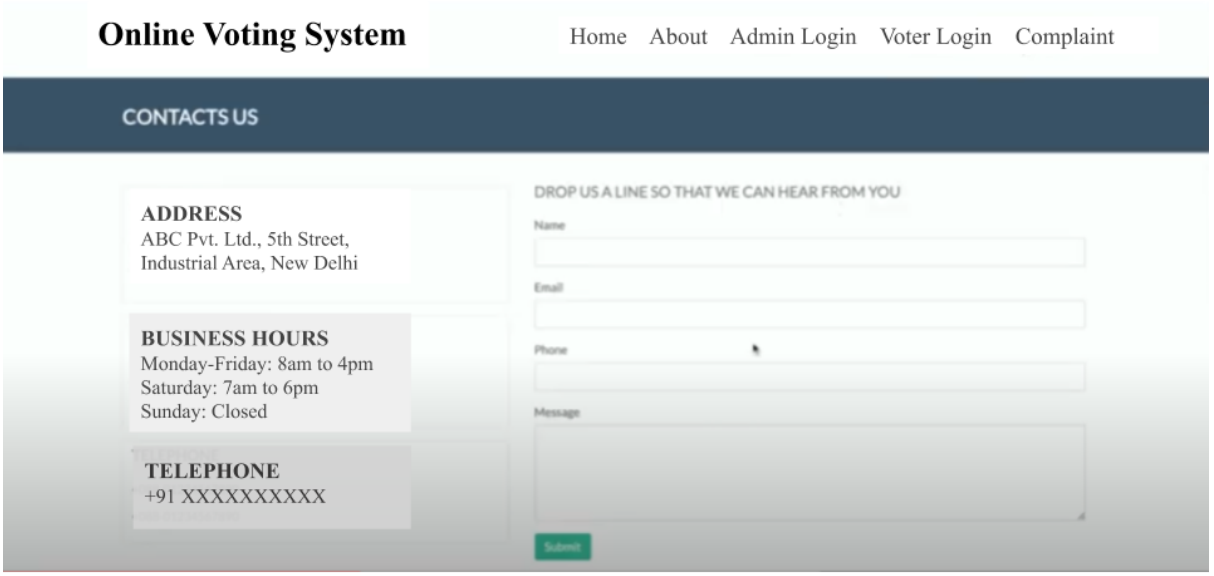


Fig 5.12 Raising an Issue and Contact Page

The final simulation of the code such that the Solenoid Lock is controlled via Fingerprint of users saved in the database is a two-step process, namely enrollment, and confirmation of saved fingerprints. Appropriate message is also shown on the LCD Display with respect to whether the fingerprint is matched or not. For the person who is registered with the system can get access through doors according to their designation. Doors opening and closing has been achieved successfully.

The software and the hardware portion of the project works together in such a way giving both the admins and users to get work done faster which takes months to compile manually and making sure everything is done correctly. The hardware also ensures security of not giving any data on the wrong hands, making the entire process safe and secure to use.

CONCLUSION & FUTURE SCOPE

This project was designed to implement a two level biometric security system for voting that will be used for the election process. The integration of biometric authentication within the system will provide an efficient way to cast votes, free of fraud, and make the system more trustable, economic and fast as well as enabling the voters to cast their votes.

The biometric fingerprint voting system provides a good solution to security. A novel architecture for an economic Fingerprint biometric technology which can be used during the electoral process is proposed and implemented. It gives a basic idea of how to detect the fingerprint of authorised voters using R305, Arduino Uno and Arduino (IDE). The cost of this technology is very economical. This project uses low cost off the shelf components, and is based on the Arduino platform which is FOSS (Free Open Source Software). So, the overall implementation cost is very cheap and is affordable by a common person.

The field of biometrics system was established and has since enlarged on to many types of annual voter recognition. Although, the person's fingerprint remains a very familiar identifier and the biometric method of options among law enforcement. These concepts of voter recognition have guided the evolution of fingerprint scanners that serve to quickly identify voters and assign access privileges. Fingerprint recognition, the electronic methods of registering and identifying an individual fingerprint, advanced significantly. Today, recognition can be achieved in a second with sensible accuracy. As a result, the use of Automated fingerprint identification systems that record, accumulate, search, test and recognize fingerprints of voters is rapidly increasing.

The security system for voting is improved by using a biometric authentication system. The main reasons for augmenting biometric authentication with a secure voting system is that biometrics are features of a person which can be hardly copied thereby it becomes very difficult to find the identity of a person . The use of fingerprint recognition deepens the process of ensuring that the voting mantra – one man, one vote – is fully enforced.

REFERENCES

[1] Brian Evans, Beginning Arduino Programming, Technology in Action, Springer, 2011.

[2] Santosh Kumar Shaw, Sashank Poddar, "Design and Implementation of Arduino Based Voting Machine", IEEE Electron Device Kolkata Conference, 24-25 November, 2018, Kolkata, India.

[3] V. Kiruthika Priya, "Arduino Based Smart Electronic Voting Machine ", ICEI, 2017.

[4] Rahul B. Diwate, Satish J. Alaspurkar, "Study of Different Algorithms for Pattern Matching", IJARCSSE, ICIS Conference, Volume 3, Issue 3, March 2013.

[5] Meena Chavan, Devanshu Saxena, "Biometric Based Electronic Voting Machine using Arduino IDE", IRJET, Volume 06, Issue 06, June 2019.

[6] Abdulkadir H. Alkali, Emmanuel G. Dada, "Design and Development of an Arduino Based Electronic Voting System, IRJES, Volume 8, Issue 1, Jan 2019.

APPENDIX A

ENTRY SIDE CODE

```
This is an example sketch for our optical Fingerprint sensor
Designed specifically to work with the Adafruit BMP085 Breakout
----> http://www.adafruit.com/products/751
These displays use TTL Serial to communicate, 2 pins are required to interface
Adafruit invests time and resources providing this open source code, please support Adafruit
and open-source hardware by purchasing products from Adafruit!
Written by Limor Fried/Ladyada for Adafruit Industries.
BSD license, all text above must be included in any redistribution
**************************************************/

#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>
#define rs 13
#define en 12
#define d7 8
#define d6 9
#define d5 10
#define d4 11
int flag=0;
// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
// uncomment this line:
// #define mySerial Serial1
// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// comment these two lines if using hardware serial
SoftwareSerialmySerial(2, 3);
LiquidCrystallcd(rs,en,d4,d5,d6,d7);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{
        pinMode(7,OUTPUT);
        digitalWrite(7,HIGH);
        Serial.begin(9600);
        lcd.begin(16,2);
        while (!Serial); // For Yun/Leo/Micro/Zero/…
        delay(100);
        Serial.println("\n\Adafruit finger detect test");
        // set the data rate for the sensor serial port
        finger.begin(57600);
        delay(5);
        if (finger.verifyPassword()) {
                Serial.println("Found fingerprint sensor!");
                lcd.clear();
                lcd.setCursor(3,0);
                lcd.print(" Welcome");
                delay(1000);
        } else {
                Serial.println("Did not find fingerprint sensor :(");
                lcd.clear();
                lcd.print("Error occurred");
                lcd.clear();
                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() {  // run over and over again
        lcd.clear();
        lcd.print("Place Finger");
        delay(100);
        getFingerprintIDez();
        delay(50); //don't need to run this at full speed.
}
uint8_t getFingerprintID() {
        uint8_t p = finger.getImage();
        switch (p) {
                case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
                case FINGERPRINT_NOFINGER:
                Serial.println("No finger detected");
                return p;
                case FINGERPRINT_PACKETRECIEVEERR:
                Serial.println("Communication error");
                return p;
                case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                return p;
                default:
                Serial.println("Unknown error");
                return p;
        } // OK success!
        p = finger.image2Tz();
        switch (p) {
                case FINGERPRINT_OK:
                Serial.println("Image converted");
                break;
                case FINGERPRINT_IMAGEMESS:
```

```
        Serial.println("Image too messy");
        return p;
        case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
        case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
        case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
        default:
        Serial.println("Unknown error");
        return p;
} // OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
        Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
        Serial.println("Communication error");
        return p;
} else if (p == FINGERPRINT_NOTFOUND) {
        Serial.println("Did not find a match");
        return p;
} else {
        Serial.println("Unknown error");
        return p;
} // found a match!
Serial.print("Found ID #");
Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);
```

```
        return finger.fingerID;
}
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
        uint8_t p = finger.getImage();
        if (p != FINGERPRINT_OK)
                return -1;
        p = finger.image2Tz();
        if (p != FINGERPRINT_OK)
                return -1;
        p = finger.fingerFastSearch();
        if (p != FINGERPRINT_OK) {
                flag++;
                Serial.println("NOT MATCH");
                lcd.clear();
                lcd.print("Not Matched");delay(1000);
                if(flag>2){
                        lcd.clear();
                        lcd.print("Unauthorised");
                        lcd.setCursor(0,1);
                        lcd.print("Access");
                        delay(2000);
                        lcd.clear();
                        Serial.println("LIMIT CROSSED");
                        //delay(500);
                        flag=0;
                }
                return -1;
        }
        digitalWrite(7,LOW);
        lcd.clear();
        lcd.print("Door Opened");
```

```
        delay(3000);
        digitalWrite(7,HIGH);
        lcd.clear();
        lcd.print("Door Closed");
        delay(100);
        lcd.clear();
        // found a match!
        Serial.print("Found ID #");
        Serial.print(finger.fingerID);
        Serial.print(" with confidence of ");
        Serial.println(finger.confidence);
        return finger.fingerID;
}
```

APPENDIX B

DATABASE CREATION

```
-- phpMyAdmin SQL Dump
-- version 4.1.14
-- http://www.phpmyadmin.net
-- Ballot Side Voting System Code
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
--
-- Table structure for table `tdadministrators`
--
CREATE TABLE IF NOT EXISTS `tbadministrators` (
`admin_id` int(5) NOT NULL AUTO_INCREMENT,
`first_name` varchar(45) NOT NULL,
`last_name` varchar(45) NOT NULL,
`email` varchar(45) NOT NULL,
`password` varchar(45) NOT NULL,
PRIMARY KEY (`admin_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
--
-- Dumping data for table `tbadministrators`
--
INSERT INTO `tbadministrators` (`admin_id`, `first_name`, `last_name`, `email`,
`password`)
VALUES
(1, 'Yashika', 'Kalra', 'y.colkalra100@gmail.com', 'voting@123');
--
-- Table structure for table `tbmembers`
--
CREATE TABLE IF NOT EXISTS `tbmembers` (
```

```
`member_id` int(5) NOT NULL AUTO_INCREMENT,
`first_name` varchar(45) NOT NULL,
`last_name` varchar(45) NOT NULL,
`email` varchar(45) NOT NULL,
`voter_id` varchar(45) NOT NULL,
`password` varchar(45) NOT NULL,
PRIMARY KEY (`member_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
--
-- Dumping data for table `tbmembers`
--
INSERT INTO `tbmembers` (`member_id`, `first_name`, `last_name`, `email`, `voter_id`,
`password`) VALUES
(1, 'Ritika', 'Garg', 'ritikagarg144@gmail.com', '19964714075000192',
'7d9b087beffc9879ead55e7291d6b541'),
(2, 'Snigdha', 'Srivastava', 'snigdhasrivastava@gmail.com', '19864714075000186',
'939d2ad38c88fda9c0bad11086e4e057'),
(3, 'Avinash', 'Kumar', ' avinashkumar@gmail.com ', '99954704075830156',
'6382gsji792gw82046h7936');
```

APPENDIX C

BALLOT SIDE CODE

HTML FILE

```
<!DOCTYPE html>
<html>
<head>
<title>online voting</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0,
user-scalable=no">
<link href="layout/styles/layout.css" rel="stylesheet" type="text/css" media="all">
<!-- <link href="css/user_styles.css" rel="stylesheet" type="text/css" /> -->
<script language="JavaScript" src="js/user.js">
</script>
</head>
<body id="top">
<div class="wrapper row0">
<div id="topbar" class="hoc clear">
<div class="fl_left">
<ul class="faico clear">
<li><a class="faicon-facebook" href="https://www.facebook.com/"><i class="fa
fa-facebook"></i></a></li>
<li><a class="faicon-pinterest" href="https://uk.pinterest.com/"><i class="fa
fa-pinterest"></i></a></li>
<li><a class="faicon-twitter" href="https://twitter.com/"><i class="fa
fa-twitter"></i></a></li>
<li><a class="faicon-dribble" href="https://dribbble.com/"><i class="fa
fa-dribbble"></i></a></li>
<li><a class="faicon-linkedin" href="https://www.linkedin.com/"><i class="fa
```

```
fa-linkedin"></i></a></li>
<li><a class="faicon-google-plus" href="https://plus.google.com/"><i class="fa
fa-google-plus"></i></a></li>
<li><a class="faicon-rss" href="https://www.rss.com/"><i class="fa fa-rss"></i></a></li>
</ul>
</div>
<div class="fl_right">
<ul class="nospace inline pushright">
<li><i class="fa fa-phone"></i> +8801773254014</li>
<li><i class="fa fa-envelope-o"></i> r.haque.249.rh@gmail.com </li>
</ul>
</div>
</div>
</div>
<div class="wrapper row1">
<header id="header" class="hoc clear">
<div id="logo" class="fl_left">
<h1><a href="index.html">ONLINE VOTING</a></h1>
</div>
<nav id="mainav" class="fl_right">
<ul class="clear">
<li class="active"><a href="index.php">Home</a></li>
<li><a href="logout.php">Logout</a></li>
</ul>
</nav>
</header>
</div>
<div class="wrapper bgded overlay"
style="background-image:url('images/demo/backgrounds/background1.jpg');">
<section id="testimonials" class="hoc container clear">
<h2 class="font-x3 uppercase btmspace-80 underlined"> Online <a
href="#">Voting</a></h2>
```

```
<ul class="nospace group">
<li class="one_half">
<blockquote>
<div >
<?php
require('connection.php');
//Process
if (isset($_POST['submit']))
{
$myFirstName = addslashes( $_POST['firstname'] ); //prevents types of SQL
injection
$myLastName = addslashes( $_POST['lastname'] ); //prevents types of SQL
injection
$myEmail = $_POST['email'];
$myPassword = $_POST['password'];
$myVoterid = $_POST['voter_id'];
$newpass = md5($myPassword); //This will make your password encrypted into
md5, a high security hash
$sql = $mysqli->query( "INSERT INTO tbMembers(first_name, last_name, email,
voter_id, password) VALUES ('$myFirstName','$myLastName', '$myEmail','$myVoterid',
'$newpass')" )
or die( mysqli_error() );
die( "You have registered for an account.<br><br>Go to <a
href=\"login.php\">Login</a>" );
}
echo "<center><h3>Register an account by filling in the needed information
below:</h3></center>";
?>
</div>
<table style="background-color:powderblue;" width="300" border="0"
align="center" cellpadding="0" cellspacing="1">
<tr>
```

```
<form name="form1" method="post" action="registeracc.php" onSubmit="return
registerValidate(this)">
<td>
<table style="background-color:powderblue;" width="100%" border="0" cellpadding="3"
cellspacing="1" >
<tr>
<td style="color:#000000"; width="120" >First Name</td>
<td style="color:#000000"; width="6">:</td>
<td style="color:#000000"; width="294"><input name="firstname" type="text" ></td>
</tr>
<tr>
<td style="color:#000000"; width="120" >Last Name</td>
<td style="color:#000000"; width="6">:</td>
<td style="color:#000000"; width="294"><input name="lastname" type="text" ></td>
</tr>
<tr>
<td style="color:#000000"; width="150" >Email</td>
<td style="color:#000000"; width="6">:</td>
<td style="color:#000000"; width="294"><input name="email" type="text" ></td>
</tr>
<tr>
<td style="color:#000000"; width="120" >Voter Id</td>
<td style="color:#000000"; width="6">:</td>
<td style="color:#000000"; width="294"><input name="voter_id" type="text" ></td>
</tr>
<tr>
<td style="color:#000000"; >Password</td>
<td style="color:#000000"; >:</td>
<td style="color:#000000"; ><input name="password" type="password" ></td>
</tr>
<tr>
<td style="color:#000000"; >Confirm Password</td>
```

```
<td style="color:#000000"; >:</td>
<td style="color:#000000"; ><input name="ConfirmPassword" type="password" ></td>
</tr>
<tr>
<td> </td>
<td> </td>
<td style="color:#000000";><input type="submit" name="submit" value="Register
Account"></td>
</tr>
</table>
</td>
</form>
</tr>
</table>
<center>
<br>Already have an account? <a href="login.php"><b>Login Here</b></a>
</center>
</blockquote>
</li>
</ul>
</section>
</div>
<div class="wrapper row4">
<footer id="footer" class="hoc clear">
<div class="one_third first">
<h6 class="title">Address</h6>
<ul class="nospace linklist contact">
<li><i class="fa fa-map-marker"></i>
<address>
<p>
Name : Md. Rezwanul Haque <br>
University : KUET <br>
```

```
Batch : 2k14 <br>
Dept : CSE <br>
</p>
</address>
</li>
</ul>
</div>
<div class="one_third">
<h6 class="title">Phone</h6>
<ul class="nospace linklist contact">
<li><i class="fa fa-phone"></i> +8801773254014<br>
+8801521479574</li>
</ul>
</div>
<div class="one_third">
<h6 class="title">Email</h6>
<ul class="nospace linklist contact">
<li><i class="fa fa-envelope-o"></i> r.haque.249.rh@gmail.com </li>
</ul>
</div>
</footer>
</div>
<div class="wrapper row5">
<div id="copyright" class="hoc clear">
<p class="fl_left">Copyright &copy; 2017 - All Rights Reserved - <a href="#">Md.
Rezwanul
Haque</a></p>
<p class="fl_right">Template by <a target="_blank" href="http://www.os-templates.com/"
title="Free Website Templates">OS Templates</a></p>
</div>
</div>
<a id="backtotop" href="#top"><i class="fa fa-chevron-up"></i></a>
```

```
<!-- JAVASCRIPTS -->
<script src="layout/scripts/jquery.min.js"></script>
<script src="layout/scripts/jquery.backtotop.js"></script>
<script src="layout/scripts/jquery.mobilemenu.js"></script>
<!-- IE9 Placeholder Support -->
<script src="layout/scripts/jquery.placeholder.min.js"></script>
<!-- / IE9 Placeholder Support -->
</body>
</html>
```

CSS FILE

```
@charset "utf-8";
/*
Template Name: Ligerrat
Author: <a href="http://www.os-templates.com/">OS Templates</a>
Author URI: http://www.os-templates.com/
Licence: Free to use under our free template licence terms
Licence URI: http://www.os-templates.com/template-terms
File: Layout CSS
*/
@import url("fontawesome-4.5.0.min.css");
@import url("framework.css");
/* Rows
-----------------------------------------------------------------------------------------------------------
*/
.row0, .row0 a{}
.row1, .row1 a{}
.row2, .row2 a{}
.row3, .row3 a{}
.row4, .row4 a{}
.row5, .row5 a{}
```

```
/* Top Bar
*/
#topbar{padding:5px 0; font-size:12px;}
#topbar .fl_right{margin-top:4px;}
#topbar .fl_right .inline > li i{margin:0 5px 0 0; line-height:normal;}
/* Header
*/
#header{padding:30px 0 0 0;}
#header #logo{margin-top:;}
#header #logo h1{margin:0; padding:0; font-size:22px; text-transform:uppercase;}
/* Page Intro
*/
#pageintro{padding:150px 0;}
#pageintro .introtxt{display:block; float:right; max-width:50%;}
#pageintro .introtxt .heading{margin-bottom:30px; font-size:42px;}
#pageintro .introtxt p{margin-bottom:30px; font-size:16px;}
/* Content Area
*/
.container{padding:80px 0;}
/* Content */
.container .content{}
/* Elements */
.elements li:last-child{margin-bottom:0;}/* Used for mobile devices when elements stack */
.elements article{display:inline-block; width:100%; max-width:320px; height:100%;}
.elements .txtwrap{padding:20px;}
.elements .txtwrap .heading{margin-bottom:0;}
.elements .txtwrap time{display:block; font-size:.8rem; margin-bottom:10px;}
.elements .txtwrap p{margin:0 0 15px 0;}
.elements .txtwrap p:last-child{margin-bottom:0;}
.elements .txtwrap footer{}
/* Comments */
#comments ul{margin:0 0 40px 0; padding:0; list-style:none;}
```

```
#comments li{margin:0 0 10px 0; padding:15px;}
#comments .avatar{float:right; margin:0 0 10px 10px; padding:3px; border:1px solid;}
#comments address{font-weight:bold;}
#comments time{font-size:smaller;}
#comments .comcont{display:block; margin:0; padding:0;}
#comments .comcont p{margin:10px 5px 10px 0; padding:0;}
#comments form{display:block; width:100%;}
#comments input, #comments textarea{width:100%; padding:10px; border:1px solid;}
#comments textarea{overflow:auto;}
#comments div{margin-bottom:15px;}
#comments input[type="submit"], #comments input[type="reset"]{display:inline-block;
width:auto;
min-width:150px; margin:0; padding:8px 5px; cursor:pointer;}
/* Sidebar */
.container .sidebar{}
.sidebar .sdb_holder{margin-bottom:50px;}
.sidebar .sdb_holder:last-child{margin-bottom:0;}
/* Testimonials
*/
#testimonials{}
#testimonials li{}
#testimonials li:last-child{margin-bottom:0;}/* Used in mobile devices */
#testimonials li blockquote{display:block; position:relative; width:100%; margin:0 0 30px 0;
padding:30px; line-height:1.4; z-index:1;}
#testimonials li blockquote::before{display:block; position:absolute; top:-15px; left:15px;
padding:0 8px; font-family:"FontAwesome"; font-size:22px; content:"\f10d";}
#testimonials li blockquote::after{display:block; position:absolute; bottom:0; left:25px;
margin-bottom:-18px; content:""; border:solid; border-width:18px 0 18px 18px;
border-color:transparent;}
#testimonials li figure{}
#testimonials li figure img{float:left; margin-right:15px;}
#testimonials li figure figcaption{float:left; margin:15px 0 0 0;}
```

```
#testimonials li figure figcaption *{display:block; margin:0; padding:0; line-height:1;}
#testimonials li figure figcaption strong{margin-bottom:5px;}
#testimonials li figure figcaption br{display:none;}
#testimonials li figure figcaption em{font-size:.8rem;}
/* Footer
*/
#footer{padding:80px 0;}
#footer .title{margin:0 0 50px 0; padding:0; font-size:1.2rem; text-transform:uppercase;}
#footer .linklist li{display:block; margin-bottom:15px; padding:0 0 15px 0;
border-bottom:1px
solid;}
#footer .linklist li:last-child{margin:0; padding:0; border:none;}
#footer .linklist li::before, #footer .linklist li::after{display:table; content:"";}
#footer .linklist li, #footer .linklist li::after{clear:both;}
#footer .contact{}
#footer .contact.linklist li, #footer .contact.linklist li:last-child{position:relative;
padding-left:40px;}
#footer .contact li *{margin:0; padding:0; line-height:1.6;}
#footer .contact li i{display:block; position:absolute; top:0; left:0; width:30px;
font-size:16px;
text-align:center;}
/* Copyright
*/
#copyright{padding:20px 0;}
#copyright *{margin:0; padding:0;}
/* Transition Fade
*/
*, *::before, *::after{transition:all .3s ease-in-out;}
#mainav form *{transition:none !important;}
/* Navigation
*/
nav ul, nav ol{margin:0; padding:0; list-style:none;}
```

```
#mainav, #breadcrumb, .sidebar nav{line-height:normal;}
#mainav .drop::after, #mainav li li .drop::after, #breadcrumb li a::after, .sidebar nav
a::after{position:absolute; font-family:"FontAwesome"; font-size:10px; line-height:10px;}
/* Top Navigation */
#mainav{}
#mainav ul.clear{margin-top:3px;}
#mainav ul{text-transform:uppercase;}
#mainav ul ul{z-index:9999; position:absolute; width:180px; text-transform:none;}
#mainav ul ul ul{left:180px; top:0;}
#mainav li{display:inline-block; position:relative; margin:0 15px 0 0; padding:0;}
#mainav li:last-child{margin-right:0;}
#mainav li li{width:100%; margin:0;}
#mainav li a{display:block; padding:0 0 32px 0;}
#mainav li li a{border:solid; border-width:0 0 1px 0;}
#mainav .drop{padding-left:15px;}
#mainav li li a, #mainav li li .drop{display:block; margin:0; padding:10px 15px;}
#mainav .drop::after, #mainav li li .drop::after{content:"\f0d7";}
#mainav .drop::after{top:5px; left:5px;}
#mainav li li .drop::after{top:15px; left:5px;}
#mainav ul ul{visibility:hidden; opacity:0;}
#mainav ul li:hover > ul{visibility:visible; opacity:1;}
#mainav form{display:none; margin:0; padding:0;}
#mainav form select, #mainav form select option{display:block; cursor:pointer;
outline:none;}
#mainav form select{width:100%; padding:5px; border:1px solid;}
#mainav form select option{margin:5px; padding:0; border:none;}
/* Breadcrumb */
#breadcrumb{padding:15px 0; text-transform:uppercase;}
#breadcrumb ul{margin:0; padding:0; list-style:none;}
#breadcrumb li{display:inline-block; margin:0 6px 0 0; padding:0;}
#breadcrumb li a{display:block; position:relative; margin:0; padding:0 12px 0 0;
font-size:12px;}
```

```
#breadcrumb li a::after{top:3px; right:0; content:"\f101";}
#breadcrumb li:last-child a{margin:0; padding:0;}
#breadcrumb li:last-child a::after{display:none;}
/* Sidebar Navigation */
.sidebar nav{display:block; width:100%;}
.sidebar nav li{margin:0 0 3px 0; padding:0;}
.sidebar nav a{display:block; position:relative; margin:0; padding:5px 10px 5px 15px;
text-decoration:none; border:solid; border-width:0 0 1px 0;}
.sidebar nav a::after{top:9px; left:5px; content:"\f101";}
.sidebar nav ul ul a{padding-left:35px;}
.sidebar nav ul ul a::after{left:25px;}
.sidebar nav ul ul ul a{padding-left:55px;}
.sidebar nav ul ul ul a::after{left:45px;}
/* Pagination */
.pagination{display:block; width:100%; text-align:center; clear:both;}
.pagination li{display:inline-block; margin:0 2px 0 0;}
.pagination li:last-child{margin-right:0;}
.pagination a, .pagination strong{display:block; padding:8px 11px; border:1px solid;
background-clip:padding-box; font-weight:normal;}
/* Back to Top */
#backtotop{z-index:999; display:inline-block; position:fixed; visibility:hidden; bottom:20px;
right:20px; width:36px; height:36px; line-height:36px; font-size:16px; text-align:center;
opacity:.2;}
#backtotop i{display:block; width:100%; height:100%; line-height:inherit;}
#backtotop.visible{visibility:visible; opacity:.5;}
#backtotop:hover{opacity:1;}
/* Tables
*/
table, th, td{border:1px solid; border-collapse:collapse; vertical-align:top;}
table, th{table-layout:auto;}
table{width:100%; margin-bottom:15px;}
th, td{padding:5px 8px;}
```

```
td{border-width:0 1px;}
/* Gallery
*/
#gallery{display:block; width:100%; margin-bottom:50px;}
#gallery figure figcaption{display:block; width:100%; clear:both;}
#gallery li{margin-bottom:30px;}
/* Font Awesome Social Icons
*/
.faico{margin:0; padding:0; list-style:none;}
.faico li{display:inline-block; float:left; margin:0 12px 0 0; padding:0; line-height:normal;}
.faico li:last-child{margin-right:0;}
.faico a{display:block; height:30px; line-height:30px; font-size:14px; text-align:center;}
.faico a{color:inherit; background-color:transparent;}
.faicon-dribble:hover{color:#EA4C89;}
.faicon-facebook:hover{color:#3B5998;}
.faicon-google-plus:hover{color:#DB4A39;}
.faicon-linkedin:hover{color:#0E76A8;}
.faicon-pinterest:hover{color:#C8232C;}
.faicon-rss:hover{color:#EE802F;}
.faicon-twitter:hover{color:#00ACEE;}
/* Colours
*/
body{color:#888888; background-color:#141414;}
a{color:#56AED4;}
a:active, a:focus{background:transparent;}/* IE10 + 11 Bugfix - prevents grey background */
hr, .borderedbox{border-color:#D7D7D7;}
label span{color:#FF0000; background-color:inherit;}
input:focus, textarea:focus, *:required:focus{border-color:#56AED4;}
.overlay{color:#FFFFFF; background-color:inherit;}
.overlay::after{color:inherit; background-color:rgba(0,0,0,.55);}
.btn, .btn.inverse:hover{color:#FFFFFF; background-color:#56AED4;
border-color:#56AED4;}
```

```
.btn:hover, .btn.inverse{color:inherit; background-color:transparent; border-color:inherit;}
.icon{color:inherit;}
.icon::before, .icon::after{background-color:#F4F4F4;}
article:hover .icon::before, article:hover .icon::after{}
.underlined::before{background-color:#D7D7D7;}
.underlined::after{color:#D7D7D7; background-color:#FFFFFF;}
.bgded .underlined::after{background-color:#000000;}
/* Rows */
.row0{color:#888888; background-color:#FFFFFF;}
.row1{color:#FFFFFF; background-color:#141414;}
.row2{color:#888888; background-color:#F4F4F4;}
.row3{color:#888888; background-color:#FFFFFF;}
.row4{color:#FFFFFF; background-color:#292929;}
.row5, .row5 a{color:#888888; background-color:#141414;}
.row2 .underlined::after{background-color:#F4F4F4;}
.row4 .underlined::after{background-color:#292929;}
/* Header */
#header #logo *{color:inherit;}
/* Content Area */
.elements article{color:#888888; background-color:#FFFFFF; box-shadow:0px 2px 5px
rgba(0,0,0,.1);}
/* Testimonials */
#testimonials li blockquote{color:#FFFFFF; background-color:rgba(41,41,41,.5);}/*
#292929 */
#testimonials li blockquote::before{color:#888888; background-color:#141414;}
#testimonials li blockquote::after{border-left-color:rgba(41,41,41,.5);}
/* Footer */
#footer .title{color:#888888;}
#footer .linklist li{border-color:#141414;}
/* Navigation */
#mainav li a{color:inherit;}
#mainav .active a, #mainav a:hover, #mainav li:hover > a{color:#5B9BBD;
```

```
background-color:inherit;}
#mainav li li a, #mainav .active li a{color:#FFFFFF; background-color:rgba(0,0,0,.6);
border-color:rgba(0,0,0,.6);}
#mainav li li:hover > a, #mainav .active .active > a{color:#FFFFFF;
background-color:#5B9BBD;}
#mainav form select{color:#FFFFFF; background-color:#141414;
border-color:rgba(136,136,136,.5);/* #888888 */}
#breadcrumb a{color:#888888; background-color:inherit;}
#breadcrumb li:last-child a{color:#56AED4;}
.container .sidebar nav a{color:inherit; border-color:#D7D7D7;}
.container .sidebar nav a:hover{color:#56AED4;}
.pagination a, .pagination strong{border-color:#D7D7D7;}
.pagination .current *{color:#FFFFFF; background-color:#56AED4;}
#backtotop{color:#FFFFFF; background-color:#56AED4;}
/* Tables + Comments */
table, th, td, #comments .avatar, #comments input, #comments
textarea{border-color:#D7D7D7;}
#comments input:focus, #comments textarea:focus, #comments
*:required:focus{border-color:#56AED4;}
th{color:#FFFFFF; background-color:#373737;}
tr, #comments li, #comments input[type="submit"], #comments
input[type="reset"]{color:inherit;
background-color:#FBFBFB;}
tr:nth-child(even), #comments li:nth-child(even){color:inherit; background-color:#F7F7F7;}
table a, #comments a{background-color:inherit;}
/* Media Queries
*/
@-ms-viewport{width:device-width;}
/* Max Wrapper Width - Laptop, Desktop etc.
*/
@media screen and (min-width:978px){
.hoc{max-width:978px;}
```

```
}
/* Mobile Devices
*/
@media screen and (max-width:900px){
.hoc{max-width:90%;}
#topbar{}
#header{padding-bottom:30px;}
#mainav{}
#mainav ul{display:none;}
#mainav form{display:block;}
#breadcrumb{}
.container{}
#comments input[type="reset"]{margin-top:10px;}
.pagination li{display:inline-block; margin:0 5px 5px 0;}
#footer{}
#copyright{}
#copyright p:first-of-type{margin-bottom:10px;}
}
@media screen and (max-width:750px){
.imgl, .imgr{display:inline-block; float:none; margin:0 0 10px 0;}
.fl_left, .fl_right{display:block; float:none;}
.one_half, .one_third, .two_third, .one_quarter, .two_quarter,
.three_quarter{display:block; float:none; width:auto; margin:0 0 30px 0; padding:0;}
#topbar{padding:10px 0 15px; text-align:center;}
#topbar .faico li{float:none;}
#topbar .fl_right ul{margin:0; line-height:normal;}
#topbar .fl_right ul li{display:inline-block; margin-right:20px;}
#topbar .fl_right ul li:last-child{margin-right:0;}
#header{text-align:center;}
#header #logo{margin-bottom:15px;}
#pageintro .introtxt{max-width:none; float:none;}
#pageintro .introtxt .heading{font-size:36px;}
```

```
#footer{padding-bottom:50px;}
}
@media screen and (max-width:450px){}
/* Other
*/
@media screen and (max-width:650px){
.scrollable{display:block; width:100%; margin:0 0 30px 0; padding:0 0 15px 0;
overflow:auto; overflow-x:scroll;}
.scrollable table{margin:0; padding:0; white-space:nowrap;}
.inline li{display:block; margin-bottom:10px;}
.pushright li{margin-right:0;}
.font-x2{font-size:1.4rem;}
.font-x3{font-size:1.6rem;}
}
```