

Faculty of Engineering and Technology Department of Electronics & Communication Engineering

Jain Global Campus, Kanakapura Taluk

Ramanagara District, Karnataka, India -562112

##### 2017-2021

**A Project Report on**

**“BIOMETRIC IDENTITY VERIFICATION SYSTEM FOR**

**NATIONAL POLLING”**

**Submitted in partial fulfilment for the award of the degree of**

BACHELOR OF TECHNOLOGY

#### IN

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted by**

#### SHAIK MOHAMMED JUNAID 17BTREC007

**ANUROOP MISHRA 17BTREC019**

**ARJUN SURESH SINGH 17BTREC020 ARPIT MALL 17BTREC021**

**Under the guidance of**

**Dr. R. Sukumar Head of the Department**

Department of Electronics & Communication Engineering Faculty of Engineering and Technology

**JAIN** (Deemed-to-be University)



Faculty of Engineering and Technology

Department of Electronics and Communication Engineering

Jain Global Campus, Kanakapura Taluk - 562112 Ramanagara District, Karnataka, India

##### 2017-2021

**A Project Report on**

### “BIOMETRIC IDENTITY VERIFICATION SYSTEM FOR

**NATIONAL POLLING”**

**Submitted in partial fulfilment for the award of the degree of**

**BACHELOR OF TECHNOLOGY IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted by**

**SHAIK MOHAMMED JUNAID 17BTREC007**

**ANUROOP MISHRA 17BTREC019**

**ARJUN SURESH SINGH 17BTREC020**

**ARPIT MALL 17BTREC021**

**Under the guidance of Dr. R. Sukumar**

Head of the Department

Department of Electronics and Communication Engineering

Faculty of Engineering & Technology

**JAIN DEEMED-TO-BE UNIVERSITY**



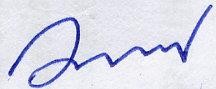
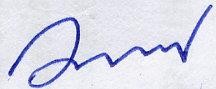
# Faculty of Engineering & Technology

Department of Electronics & Communication Engineering

Jain Global campus Kanakapura Taluk - 562112 Ramanagara District Karnataka, India

# CERTIFICATE

This is to certify that the project work titled **“BIOMETRIC IDENTITY VERIFICATION SYSTEM FOR NATIONAL POLLING”** is carried out by **Shaik Mohammed Junaid (17BTREC007), Anuroop Mishra (17BTREC019), Arjun Suresh Singh (17BTREC020), Arpit Mall (17BTREC021),** are bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY, Bengaluru in partial fulfillment for the award of degree in Bachelor of Technology in Electronics and Communication Engineering, during the year **2020-2021**.



##### Dr. R. Sukumar Dr. R. Sukumar Dr. Hariprasad S. A

Head of the Department Dept. of ECE,

Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY

Date:

Head of the Department, Electronics and Communication, Faculty of Engineering & Technology,

JAIN DEEMED-TO-BE UNIVERSITY

Date:

Director,

Faculty of Engineering & Technology,

JAIN DEEMED-TO-BE UNIVERSITY

Date:

Name of the Examiner Signature of Examiner 1.

2.

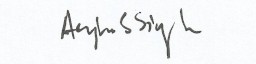
# DECLARATION

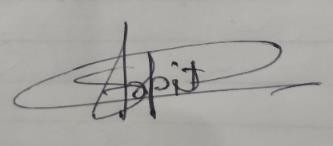
We, **Shaik Mohammed Junaid (17BTREC007), Anuroop Mishra (17BTREC019), Arjun Suresh Singh (17BTREC020), Arpit Mall (17BTREC021),** are students of eighth semester B-Tech in **Electronics and Communication Engineering**, at Faculty of Engineering & Technology, **JAIN DEEMED-TO-BE UNIVERSITY**, hereby declare that the project titled **“Biometric Identity Verification System for National Polling”** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Electronics and Communication Engineering** during the academic year **2020-2021**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

Shaik Mohammed Junaid 17BTREC007

Anuroop Mishra 17BTREC019

Arjun Suresh Singh 17BTREC020

Arpit Mall 17BTREC021

Place: Bengaluru Date: 20/05/2021

## ACKNOWLEDGEMENT

*It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.*

*First, we take this opportunity to express our sincere gratitude to* **Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY** *for providing us with a great opportunity to pursue our Bachelor’s Degree in this institution.*

*In particular we would like to thank* ***Dr. Hariprasad S.A****,* ***Director****,* **Faculty of Engineering & Technology**, **JAIN DEEMED-TO-BE UNIVERSITY** *for his constant encouragement and expert advice.*

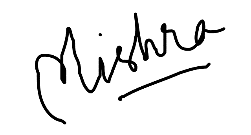
*It is a matter of immense pleasure to express our sincere thanks to* ***Dr. R. Sukumar****,* **Head of the department**, **Electronics and communication Engineering**, **JAIN DEEMED- TO-BE UNIVERSITY*,*** *for providing right academic guidance that made our task possible.*

*We would like to thank our guide* ***Dr. R. Sukumar****,* **Head of the department***,* **Dept. of Electronics and Communication Engineering**, **JAIN DEEMED-TO-BE UNIVERSITY***, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

*We would like to thank our Project Coordinator* ***Mr. Sunil M P*** *and all the staff members of Electronics and Communication for their support.*

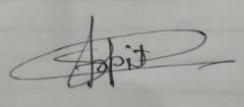
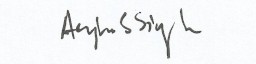
*We are also grateful to our family and friends who provided us with every requirement throughout the course.*

*We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.*

*Signature of Students*



*Shaik Mohammed Junaid Anuroop Mishra*



*Arjun Suresh Singh Arpit Mall*

# ABSTRACT

The project “Biometric Identity Verification System for National Polling” intends to provide a reliable and efficient solution to the problems faced in verification at the time of elections. Keeping in mind the scale of elections in our country, this is a much-needed step in order to increase the efficiency of the system. Presently, the procedure depends mostly on manual checking which is paper-based and hence less secure and efficient. This type of mechanism can lead to wastage of time and efforts at a gigantic scale.

Thus, to eliminate these flaws and to improve the overall all performance of the process, this project aims to utilize the ease of use of modern technologies like IoT and Raspberry Pi, along with the security and reliability of biometric based verification system.

IoT serves as the medium of communication through which we configure the hardware sensors to send and receive data to and from the cloud-based storage. The cloud storage used in this project is “Firebase” by Google which is not only fast and efficient, but also cost-effective. All the sensors are connected to the Raspberry Pi, which is supplied with the internet. Hence all the data scanned from the sensors is transferred to the cloud over the internet using Raspberry Pi and the commands sent by cloud in return are then given as results.

This type of system does not require a need for physical database at all times and locations which makes it ideal for use all over the country. The basic requirements of the model are power supply and reliable internet.

**Keywords:** Internet of Things (IoT); Biometrics; Firebase; Cloud Storage; Raspberry Pi; Fingerprint Sensors; Polling.

## TABLE OF CONTENTS

[List of Figures viii](#_TOC_250020)

[Nomenclature used x](#_TOC_250019)

[Chapter 1 11](#_TOC_250018)

1. [INTRODUCTION 11](#_TOC_250017)
   1. [Literature Survey 14](#_TOC_250016)
   2. [Limitations of the Current Work 15](#_TOC_250015)
   3. [Problem Definition 15](#_TOC_250014)
   4. Objectives 16
   5. [Methodology 17](#_TOC_250013)
   6. Hardware and Software tools used 18

[Chapter 2 19](#_TOC_250012)

1. [BASIC THEORY 19](#_TOC_250011)

[Chapter 3 26](#_TOC_250010)

1. [TOOL DESCRIPTION 26](#_TOC_250009)

[Chapter 4 41](#_TOC_250008)

1. [IMPLEMENTATION 41](#_TOC_250007)
   1. Hardware Design and Implementation 41
      1. [Fingerprint Module 41](#_TOC_250006)
      2. [Raspberry pi 43](#_TOC_250005)
      3. [Database 44](#_TOC_250004)
   2. Software algorithm 50

[Chapter 5 60](#_TOC_250003)

1. RESULTS AND DISCUSSION 60

CONCLUSIONS AND FUTURE SCOPE 66

[REFERENCES 68](#_TOC_250002)

APPENDICES

APPENDIX – I 69

DETAILS OF PAPER PUBLICATION (ALONG WITH PAPER)

[INFORMATION REGARDING STUDENTS 83](#_TOC_250001)

[BATCH PHOTOGRAPH ALONG WITH GUIDE 84](#_TOC_250000)

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Fig. No. | Description of the figure | Page No. |
| Fig. 1.1 | Flowchart of methodology | 18 |
| Fig. 2.1 | Operation of a biometric system | 20 |
| Fig. 2.2 | FAR-FRR Graph | 21 |
| Fig. 3.1 | Raspberry Pi 3 B+ | 26 |
| Fig 3.2 | Pinout of Raspberry Pi 3 B+ | 28 |
| Fig. 3.3 | Dimension of Raspberry Pi 3 B+ | 30 |
| Fig. 3.4 | Inbuilt Wi-Fi Adaptor in Raspberry Pi 3 B+ | 31 |
| Fig. 3.5 | Inbuilt Wi-Fi adapter working | 32 |
| Fig. 3.6 | R307 Fingerprint Scanner | 33 |
| Fig. 3.7 | Working of Fingerprint Scanner | 34 |
| Fig. 3.8 | Male to Male Jumper Wires | 35 |
| Fig. 3.9 | Female to Female Jumper Wires | 35 |
| Fig. 3.10 | Male to Female Jumper Wires | 35 |
| Fig.3.11. | Setting up Raspbian OS in Linux | 37 |
| Fig. 4.1. | System Architecture | 41 |
| Fig. 4.2 | Fingerprint Sensor R307 | 41 |
| Fig. 4.3. | Serial connection to the back of fingerprint module | 42 |
| Fig. 4.4. | USB to Serial TTL UART Converter | 43 |
| Fig. 4.5 | Connection of the board to converter | 44 |
| Fig. 4.6. | Firebase Realtime Database | 44 |
| Fig. 4.7 | Landing Page of Firebase | 45 |
| Fig. 4.8. | Console page | 45 |
| Fig. 4.9 | Entering the project name | 46 |
| Fig. 4.10 | Configuring Google Analytics | 46 |
| Fig.4.11. | Click on “Create Project” after all the configuring | 47 |
| Fig.4.12. | Overview of our project | 47 |
| Fig. 4.13 | Setting database location | 48 |
| Fig. 4.14 | Selecting mode of databases | 48 |
| Fig. 4.15 | Working of Fingerprint Sensor | 49 |

|  |  |  |
| --- | --- | --- |
| Fig. 4.16 | Raspberry Pi 3 Model B+ | 49 |
| Fig. 4.17 | Main Function | 51 |
| Fig. 4.18 | Code for Registration | 52 |
| Fig. 4.19 | Fingerprint code to interface database and scanner | 53 |
| Fig. 4.20 | Data Push Code | 54 |
| Fig.4.21. | Validation Code | 55 |
| Fig.4.22. | Fingerprint Store Driver Code (part 1) | 56 |
| Fig. 4.23 | Fingerprint Store Driver Code (part 2) | 57 |
| Fig. 4.24 | Fingerprint load driver code (part 1) | 58 |
| Fig. 4.25 | Fingerprint load driver code part 2 | 59 |
| Fig 5.1 | Fingerprint Sensor connected to Raspberry Pi | 60 |
| Fig 5.2 | Empty database right after set-up | 61 |
| Fig 5.3 | Registration process for first user | 62 |
| Fig 5.4 | Database after pushing data for first user | 63 |
| Fig 5.5 | Registration process for second user | 63 |
| Fig 5.6 | Database after pushing data for second user | 64 |
| Fig. 5.7 | Validation Process | 65 |
| Fig. 5.8 | Final result for validation | 66 |

**NOMENCLATURE USED**

|  |  |
| --- | --- |
| IoT | Internet of Things |
| EVM | Electronic Voting Machine |
| ID | Identity |
| AES | Advanced Encryption Standard |
| USB | Universal Serial Bus |
| OS | Operating System |
| FRR | False Rejection Rate |
| FAR | False Acceptance Rate |
| EER | Equal Error Rate |
| CPU | Central Processing Unit |
| GPU | Graphical Processing Unit |
| SoC | System on Chip |
| RAM | Random Access Memory |
| PoE | Power over Ethernet |
| HDMI | High-Definition Multimedia Interface |
| GUI | Graphical User Interface |
| GPIO | General-Purpose Input/Output |
| TTL | Transistor-Transistor Logic |
| UART | Universal Asynchronous Receiver/Transmitter |

# Chapter 1

# INTRODUCTION

Consensus forms the backbone of any democracy. It is essential for the efficient working of a nation and its government. However, there has been little to no technological advancements lately in the polling mechanism regardless of how crucial it is to digitize the procedure with the emerging technologies. One such technology that can be used to ameliorate the process is the Internet of Things or simply known as IoT. IoT refers to a network where physical devices are connected to the internet for communication. Connected with cloud, IoT can solve major problems faced during national polling and can serve as a reliable alternative for conventional methods.

Best served by a group of secure data points; IoT relies on the integrity of the data sent and received. Those data points share vital information and make important connections that establish relationships and recommendations. Sensitive user data is often contained in those recommendations and hence the security and reliability of the biometrics plays the important role in strong security of the connected devices and retained data. Thus, we enable biometric security at various points in the IoT data pathway. For example: - Using fingerprint sensor to secure a device and use of smartcard for identity verification. These security points help create a seamless experience and allow data to flow freely and quickly between points. That data arriving untampered is important. This allows for quick and efficient connections to be made, and keeps the data used to make those connections as secure as possible during the transfer and delivery processes.

Biometrics provide a secure way to transfer data as well as identify data ports and devices and ensure that they remain secure and their data intact. Biometrics are an optimal security measure, and their continued development will be a key component to creating difficult to breach security protocols. Since the characteristics identified by biometric scanners do not change and are unique to each individual, they make a very secure means of communicating data and creating identifiers for sharing secured data.

No matter how strong the encryptions or passwords are, both can be cracked or breached with some skills. While, it is close to impossible to duplicate or forge the biometrics. This feature alone makes them worthy of consideration in any security system.

The biometrics industry will continue growing at a fast pace, and hence biometrics will continue to make way into security systems of all levels of technologies. With a need to keep data secure and the ease of integration into a variety of systems, the technology will continue to expand and will help develop seamless data transfer that is as secure as possible. Making the connections that are important to the IoT and the recommendations that end users have come to rely on, the role of biometrics will grow and evolve along with the IoT. The unique nature of biometrics and the myriad of ways that they can be used is likely to play a pivotal role in the growth of the IoT.

Three major challenges faced during elections: Need of manual labor, Extensive paperwork involved, Lack of automation, Prevention of frauds and tempering. There has been a lot of concerns raised regarding these challenges which develop the need for automation. Most of the aforementioned problems can be solved to a large extent by automating the process. The process can be divided into three for a better understanding of what the project deals with. First is the identity verification part where conventional methods use “Voter ID” cards. Second is the voting part which is achieved currently using EVM’s. Lastly, it deals with confirmation of votes and prevention the casting of multiple votes. This uses an even traditional approach with the use of inks and papers.

Biometrics can basically be used for two main purposes:

* + For prevention of registering the same person multiple times.
  + For identification of the person registering.

There are more than one reasons to why people register multiple times. It can be to create an opportunity to be able to vote several times, it can be to obtain other services several times (e.g., pensions), it can be just for the kicks of being able to cheat the system, or it can simply be due to some kind of misunderstanding. A common occurrence of re-registration is during re-location. Since the system in general is not designed to allow the change of address, registering again at this time without getting the old records deleted causes multiple entries. Any registration process design should keep in mind sustainability and the long-term maintenance of the system. The use of biometrics for identification on election

day is not a common motivation for civil/voter registration. There are no cases known to this author of countries employing biometrics on election day. Card readers or barcodes are used to check the eligibility of a voter at the particular polling location in a few countries, but these do not include biometric checks. This method is only to eliminate the use of paper list as finding in these lists if more time taking and inefficient. The check that the ID card belongs to the person holding it is usually a simple visual check that the face of the person corresponds to the photo on the ID card.

In this project, our team intends to solve these challenges by integrating the first and the last part of the voting mechanism and implementing it using IoT. This would not only digitize the complete mechanism but also strengthen it by providing reliable solutions to the problems faced currently. Biometrics, as we all know, is the best way to verify a person’s identity. Using biometrics will make the process more secure and is easily achievable.

The biometrics of the voter is stored in the cloud which can be accessed at all times and locations with ease using the internet and Raspberry Pi. This automated approach solves all the problems to a large extent. It alleviates the painful process that involves manual labor and makes the overall structure for verification more seamless and efficient.

## Literature Survey

To understand the need of the said model and the amount of progress already made in this field, we went through a lot of pre-existing works. These works in the given field were a great motivation to give an idea of the current state of the model with all its pros and flaws. Few of them are stated below along with their advantages and disadvantages:

* + - **1.** “Implementation of Blockchain for Fair Polling System” by Anubhav Mishra, Anuroop Mishra, Abhyudya Bajpai and Abhinav Mishra.

##### Advantages:

**i.)** Lay’s layout of a complete digital voting mechanism.

**ii.)** Describes all the technologies that can be used for this in detail

**iii.)** Presents improved alternative for all three parts of polling: Identification; Vote Casting; Authentication.

##### Limitation:

**i.)** Presents only a broader overlook of digital identification and authentication with more focus on the vote casting part.

**ii)** Doesn’t discuss all the deeper aspects of all the hardware and software components.

* + - **2.** “IoT Based Biometric Implementation on Raspberry Pi” by Dhvani Shah and Vinayak Haradi.

##### Advantages:

**i.)** Gives profound knowledge of all the components and requirements.

**ii.)** Provides outline of complete mechanism.

**iii.)** Strengthens security of the model by using AES-256 encryption techniques.

##### Limitation:

**i.)** Depends totally upon a single biometric trait i.e., fingerprint.

**ii.)** Lack of two-factor authentication raises security concerns.

* + - **3.** “A Novel Design of Electronic Voting System Using Fingerprint” by Ashok Kumar D and Ummal Sariba Begum T.

##### Advantages:

i.) Uses biometrics for identification and explains the working of process clearly.

##### Limitation:

i.) Lack of two-factor authentication raises security concerns.

ii.) Does not uses any technology like IoT which raises questions about availability of the storage database at all locations.

## Limitations of the Current Work

The election is an important part of any democracy, and it’s important to ensure that this process is made as seamless as possible, both for the voters and the people involved in setting up the process. There are certain problems with our conventional systems, some of which are discussed below

* One of the biggest problems with the conventional system is that they are not biometric-enabled, and thus leave room for spoofing identity, as well as potentially recasting votes.
* Requires manual checking of the recasting of votes.
* Requires manual verification of identity involving paperwork etc.
* All of these manual works are prone to the occasional human error.
* Increased time consumption leads to large queues and unnecessary delays.
* Paperwork and manual checking restrict the polling location for each voter.
* All the inconvenience and polling location restrictions result in less voting percentages overall in elections

The aim of this project is to deal with potential issues concerning identification, eligibility, and to remove as much human intervention as possible.

## Problem Definition

Due to being the largest democracy in the world, the scale of voting in India is colossal. Thus, minuscule issues at low level amalgamate to being bigger problems when considered for the whole country. Currently, the voting system in India deals with these issues mostly due to lack of automation in the process. Even automating a few parts of the complete

system can cause drastic changes and improve the reliability and efficiency of the mechanism.

In the conventional method, at the time of elections, each voter is allotted with a polling booth with the help of his/her voter id. The polling officials at the booth have the necessary data of each voter for their polling booth. On the day of the election, the voter who enters the booth to cast the vote goes through a manual identity check. After the vote is cast, the polling official marks that the specific voter has cast the vote on paper and uses ink to mark the voter and prevent him/her from casting the vote again. The complete process requires a large amount of time, paperwork and human efforts. On top of that, since the process is manual, there are high chances of errors and frauds. Considering the gravity of the matter, possibility of any such error needs to be eliminated with the use of a more secure structure.

All this can be dealt with ease with the help of IoT. Internet of Things is more reliable, secure, time-saving, requires less human involvement and if environment-friendly as well thanks to the removal of paperwork. Also, IoT makes it possible for a voter to cast his vote from anywhere in the country and not being restricted to a single polling booth. A voter can go reach any polling booth accessible to him and cast his/her vote. This will increase the voting percentages significantly. In a bigger picture, all this amounts to stupendous improvements in the procedure by solving all the major problems encountered.

## Objective and scope of the project

The objective of this project is to deal with current potential issues concerning identification, eligibility, and human intervention in the process of polling. A biometric- enabled authentication deals with accurate identification of voter, with reference to a citizenship database containing their biometric credentials, thus preventing identity spoofing. When connected to a database, this also helps prevent the casting of multiple votes by the same voter. Thus, the biometric aspect of this system assures genuine votes.

Moreover, an IoT enabled system enables us to host a citizenship database with relevant information on the citizens, which holds their biometric traits as the primary identification factor. Having it IoT empowered also enables the update of relevant information related to authentication and scrutiny in real-time to a cloud database. Having it IoT enabled makes

this whole process seamless with minimal human intervention, thus decreasing possibilities of error, bias, and reducing the time taken in the entire process.

These advancements will make the entire voting process more seamless than ever, and massively reduce the cost, time, and effort that goes into setting up our polling system, considering the scale of the polling process on the account of the population of our country. More importantly, these advancements will guarantee, to a much greater degree than before, the genuineness of the cast votes.

## Methodology

Moving on the methodology of the model, this flowchart below gives an overview of the work flow. Each step is explained below:

* + - As the voter enters, he/she is asked to input the biometric traits (fingerprints/facial/iris).
    - These traits are scanned by the sensors which sends the scanned data to the Raspberry Pi.
    - Raspberry Pi, over the internet, sends these scanned data to the cloud where the traits are matched with the database of registered users.
    - If the eligibility criteria are met, the system approves the user for proceeding to vote.
    - If the person is not enrolled or ineligible to vote, the permission to vote is denied.

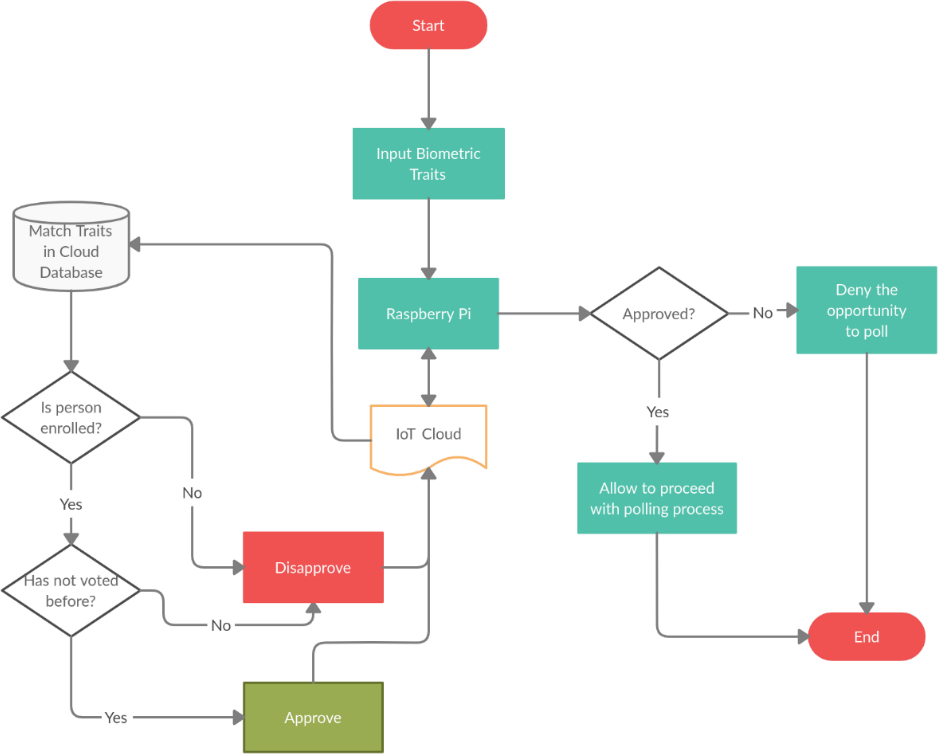


Fig. 1.1. Flowchart of methodology

## Hardware & Software Tools to be used

##### Hardware:

* Raspberry Pi 3 B+
* USB Fingerprint Scanner

##### Software:

* Raspbian OS
* Cloud storage platform
* Python

# Chapter 2

### BASIC THEORY

Elections are a defining feature of democratic government, but all too frequently, we take the actual mechanics of the election for granted. We speak at length of such issues as who is allowed to vote, how campaigns are conducted, and how they are financed, but no one gives priority to the understanding of the actual voting process.

The EVMs reduce the voting and counting time compared to the traditional method i.e., old paper ballot system. So, our main agenda is to check voter is eligible or not with the help of biometric before going to electronic voting machine (EVM) to give their vote. Identification and validation of user is done by manually through polling officer.

Now a day’s our election commission uses traditional system which includes following steps. When voter enter in polling station, Election officer search his name in register and take signature in register. After identification is done by first officer, second officer will apply indelible ink usually on left index finger of the voter. Then the officer gives you a slip that bears the voter register number where you signed or put your thumb impression. Voter can hand over this slip to the presiding officer he can confirms the serial number and permits for vote by pressing the button of the control unit of EVM. These methods having drawbacks which are user identification the main issue of cheating for duplication and rigging. To overcome this problem, we are using IoT based biometric identification system.

The main components of the system for the purposes of this discussion are the capture (whereby the sensor collects biometric data from the subject to be recognized), the reference database (where previously enrolled subjects’ biometric data are held), the matcher (which compares presented data to reference data in order to make a recognition decision), and the action (whereby the system recognition decision is revealed and actions are undertaken based on that decision.

This diagram presents a very simplified view of the overall system. The operational efficacy of a biometric system depends not only on its technical components—the biometric sample capture devices (sensors) and the mathematical algorithms that create and compare

references—but also on the end-to-end application design, the environment in which the biometric sensor operates, and any conditions that impact the behavior of the data subjects, that is, persons with the potential to be sensed.

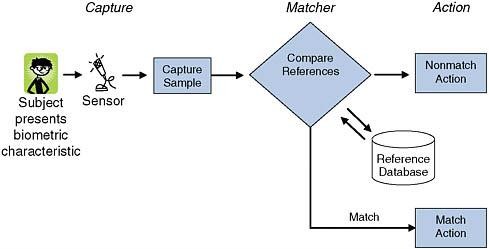


Fig. 2.1. Operation of a biometric system

* 1. **Functions**

A biometric system can provide the following two functions:

* + - **Verification (‘One-to-One’** or **1:1)** - Authenticates its users in conjunction with a smart card, username or ID number. The biometric template captured is compared with that stored against the registered user either on a smart card or database for verification.
    - **Identification (‘One-to-Many’** or **1:N)** - Authenticates its users from the biometric characteristic alone without the use of smart cards, usernames or ID numbers. The biometric template is compared to all records within the database and a closest match score is returned. The closest match within the allowed threshold is deemed the individual and authenticated.

## Performance

Biometric systems are susceptible to the following kinds of errors:

* + - **FRR (False Rejection Rate)** - the percentage of identification instances in which unauthorized persons are incorrectly accepted.
    - **FAR (False Acceptance Rate**) - the percentage of identification instances in which authorized persons are incorrectly rejected.

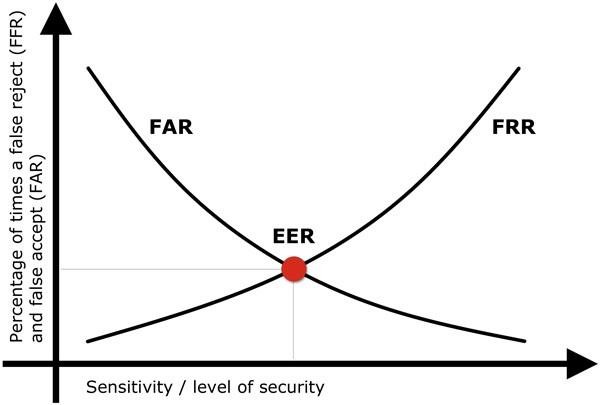


Fig. 2.2. FAR-FRR Graph

* + - To evaluate the global percentage of error of the system is used the EER (Equal Error Rate), defined as the percentage when the FAR and FRR are equal.

## The Internet of Things (IoT): An Overview

The Internet of Things (IoT) refers to devices connected to each other and to a device or app that can control them. These devices may be connected through the Internet, Bluetooth, or other means. While connecting these devices can be highly beneficial, it can also provide another invasive opportunity to threaten, harm, monitor or harass a survivor. On the contrary, it can also be used to offer potential solutions for the survivors to increase their safety strategically.

In the aspects of technical, social, and economic significance, Internet of Things is an emerging technology. All kinds of consumer products, goods, cars, trucks, appliances and all other industrial and utility components like sensors and actuators can be connected together in a network with the help of Internet of Things which along with the data analytic capabilities, is transforming our way of working, living and playing. There are several impressive projections for the impact of Internet of Things on the economy with some even anticipating as many as 100 billion devices connected through IoT and therefore a global economic impact of more than $11 Trillion by 2025.

Hacking of IoT-connected devices, security and surveillance concerns, and privacy fears have been attention-grabbing headlines about the IoT which has caused the concerns about the significant challenges faced by IoT, to rise drastically in the recent times. These challenges can hinder the way of potential benefits IoT poses.

#### IoT Characteristics

Due to integration of sensors and other utility components that can conduct complex data analytics, IoT devices are often referred as “smart” devices. The data is collected by IoT devices through the sensors and various services are offered based on the results of different analyses on that data in accordance with the user-defined parameters. Example: Cameras are the sensors in the case of smart refrigerators and these can be used to maintain the inventory and alert the user when a particular item is running low in the stock based on image recognition. Several IoT devices can “learn” by finding patterns in the user preferences and earlier use of data. An IoT device can be made “smarter” as its program is adjusted to increase the efficiency of its prediction capability which can be used to enhance the user experience or utility.

IoT devices can be connected to the internet either directly, or indirectly through another IoT device, or using both. Network connections are used for sending and receiving information and communication with users. The IoT creates bridges and connections between the IoT devices with the means of amalgamating software applications. IoT devices can make it possible for the end users to gain access to the information or control devices from any location with the help of various internet-connected devices. For example, a smart doorbell and lock may allow a user to see and interact with the person at the door and unlock the door from anywhere using a mobile device or computer.

#### IoT Application

IoT devices are used in different fields for a broad range of functions. This section below describes several categories of IoT.

* + - * **Industrial Internet of Things (IIoT):** The production and manufacturing industry has begun to adopt conventional IoT applications. Referred to as industrial Internet of Things (IIoT), machines of a production facility that are connected to the network can communicate as well as share information to improve the productivity of goods

and efficiency of the system. The use cases of IIoT can vary greatly, from spotting corrosion on the inner walls of a refinery pipe to delivering real-time data of the production value. Currently in North America, the number of consumers of IIoT connections are far less than that of IoT connection, but this can change in the future. IIoT has the capability to remodel a vast number of industries, including manufacturing, textile, food and beverage, automotive, and steel. The integration of IIoT and analytics is seen by the industry pundits as the Fourth Industrial Revolution.

* + - * **Internet of Medical Things (IoMT):** Internet of Medical Things is created by the incorporation of IoT in the healthcare field. Heart monitor, pace maker and other devices of such kind can collect and share the statistics of patients’ health over the internet to various healthcare providers to monitor, analyze and configure remotely. At a personal level of health monitoring, IoT devices that are wearable, like fitness trackers and watches, can be used to track the physical activities of a user, basic vital data, and sleeping patterns. Nearly one in five Americans use a smart watch or a fitness tracker, states a 2019 survey conducted by Pew Research.
      * **Smart Cities:** IoT devices and systems in the utilities, transportation, and infrastructure sectors may be grouped under the category of “smart city.” Utilities can used to create “smart” grids of IoT networks along with the electricity meters, water meters, and gas meters, where the sensors collect, send and receive data of customer usage. This data is used to make it possible for the central control system to optimize manufacturing and distribution to meet real time demands. For fare reading, status tracking or locating that interface with the public transportation platforms can be done using transportation IoT. For example, Columbus, Ohio’s leading proposal for the Department of Transportation’s 2016 Smart City Challenge integrated connected infrastructure which interacts with the vehicles and common payment and trip planning system across multiple transit systems.
      * **Smart Homes:** IoT devices meant for consumers that are used in the homes and other buildings are grouped under the “smart home” domain, including smart TVs, appliances, infotainment systems, light bulbs, door locks, home security system, outlets and bells. The smart home devices that use IoT can be connected to a unique network and controlled remotely via a mobile device or computer over the internet.

#### Advantages of IoT

IoT facilities several advantages for a day-to-day life in business sector. Few of those these benefits are:

* **Efficient resource utilization:** We can drastically increase the efficiency of resource utilization once we know the particular functionality of how each device works and monitor natural resources.
* **Minimize human effort:** Once the devices are connected together in an IoT network, they can interact with one another and do a lot of jobs on their own minimizing the human intervention and efforts.
* **Save time:** Thus, a lack of need for the human involvement and command registration, a lot of time can be saved using IoT devices. Time is the primary factor which can be saved with the use of IoT networks.
* **Enhance Data Collection:** Smart devices when working collectively can efficiently collect the data needed and analyze it.
* **Improved Security:** Now, one of the main benefits of the IoT devices is that when incorporated properly, it can improve the system security and reliability.

## Biometric Security in IoT

Biometric security or biometric recognition considers two criteria about a user’s body; Characteristics and uniqueness. Few of the most prominent biometric characteristics that are used in IoT biometric security are facial recognition, fingerprint recognition, iris recognition, palm geometry, DNA, etc. The biometric is generally chosen based upon the requirements of the authentication usage. For example, smartphone devices are best implemented with the voice biometrics because the microphones are already embedded in the mobile phones, and the most intriguing part of this type of biometric authentication is that it can recognize the user which is not already registered within the system, but is still trying to get the access. There are two types of biometrics modalities:

i.) Physiological

ii.) Behavioral

#### Biometric Security System

The problem of identity matching can be categorized into 2 different types with different complexity: (a) authentication (b) identification. Authentication check the eligibility of the claim of a person, whereas identification process is used to recognize the identity. Although, the reliability and ease of incorporation of a fingerprint-based authentication are far more than the signature, signature is still the more commonly used type of biometric authentication.

#### Benefits in Using Biometric-Based Security Schemes for IoT Applications

Biometric-based security can be used in every application are of IoT. Biometric issued in the application level where man and machine interaction are required.

* Security systems can be usable in smart home systems.
* A person can use smart lock system in the door using biometric locking system.
* An IoT healthcare platform that is enabled by biometric identity verification can be used by an old person to easily login and report his/ her health condition.
* In the IoT of transportation systems, the system can verify the identity of an end user while parking a car or paying traffic fine etc. Traffic police can verify whether a car belongs to a driver or not using a biometric verification means.
* IoT healthcare systems can also utilize biometric identity verification for recognition of all the medical personals before prescribing using biometrics.

# Chapter 3

### TOOL DESCRIPTION

* 1. **Hardware**

Basic overview of the hardware components used in our project are discussed below:

#### Raspberry Pi 3 B+

* + - * Raspberry Pi foundation first launched Raspberry Pi 3 B+ on 14th March 2018 making it the advanced successor of Raspberry Pi 3 B model that was introduced in 2016.
      * Raspberry Pi 3 B+ has the capability of doing some functions of a regular computer due to the presence of USB Ports, GPU, CPU, I/O pins, Bluetooth, Wi-Fi and network boot capabilities.
      * Features of the B+ version is almost same as B model; however, [USB](https://www.theengineeringprojects.com/2018/11/introduction-to-usb.html) and Network Boot and Power over Ethernet facility only come with B+ model. Also, two extra USB ports are added to this device.
      * The SoC incorporates both CPU and GPU on a single package which makes it faster than Pi 2 and Pi 3 models.

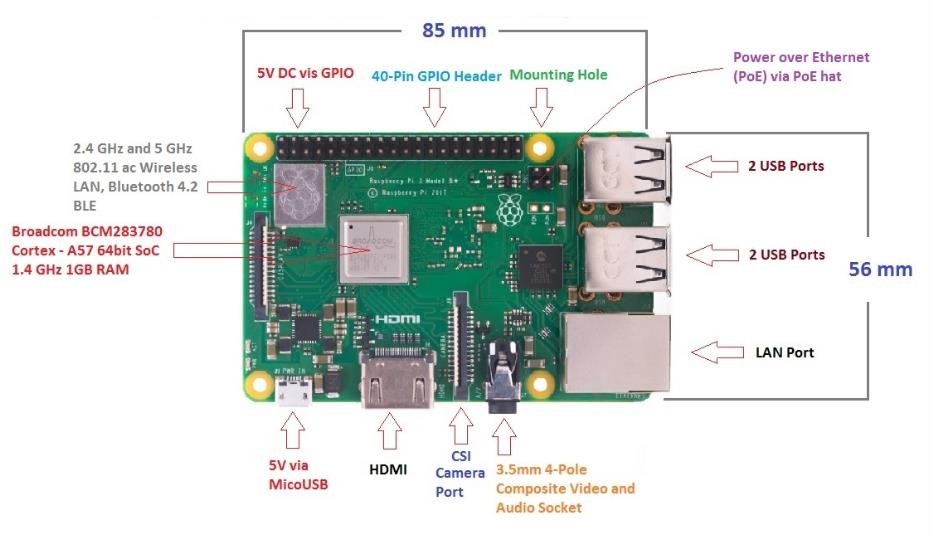


Fig. 3.1. Raspberry Pi 3 B+

##### Improvement over the Raspberry Pi 3

* + - * The model B+ is in lead in terms of processing power and has an improved wireless capability.
      * The dual-band Wi-Fi 802.11ac runs at 2.4GHz and 5GHz and provides a better range in wireless challenging environments and Bluetooth 4.2 is available with BLE support.
      * The top side is now painted with a metal shielding, replacing the plastic shielding in the earlier models, which acts as a heat sink and drains the excessive heat if the board is subjected to the high temperature or pressure.
      * This B+ model is three times faster than Pi 2 and 3 which is a major development in terms of speed, capable of executing different functions at a decent pace.
      * Exceeding the earlier version that had 100 Mbit/s speed, the ethernet port in 3 B+ comes with 300Mbit/s. It is known as gigabit ethernet based on USB 2.0 interface.
      * Due to the 4 pin headers that are added on the new board near the 40 pin headers, it now allows Power over Ethernet (PoE) which provides the required electrical current to the device with the help of data cables instead of power cords. It is very useful and reduces the number of cables required for the installation of a device in the relevant project.
      * PoE works only in the presence of PoE hat.

##### Raspberry Pi 3 B+ Pinout

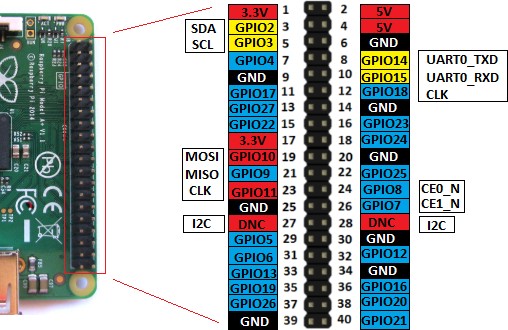


Figure 3.2 Pinout of Raspberry Pi 3 B+

* + - * + The external connection with the electronic device is developed utilizing 40 pin headers. This is the same as the previous versions, making it compatible with all the devices where older versions can be used.
        + Out of 40 pins, 26 are used as a digital I/O pins and 9 of the remaining 14 pins are termed as dedicated I/O pins which indicate they don't come with alternative function.
        + Pin 3 and 5 comes with an onboard pull up resistor which 1.8 KΩ and Pin 27 and 28 are dedicated to ID EEPROM. More space is allowed in B+ for the additional mounting hole by repositioning the GPIO header. The devices that are compatible with the B model may work with the B+ version; however, they may not sit identically to the previous version.

##### Hardware Specifications

1. **CPU**: The CPU comes with a capacity of 64 bit. CPU acts as the brain of the Raspberry Pi and helps in performing a number of instructions based on the various mathematical and logical formulas.
2. **Clock Speed and RAM:** CPU comes with a clock speed of 1.4 GHz Broadcom BCM2837B0 and consists of quad-core ARM Cortex-A53. RAM memory is identical to the previous versions being around 1GB.
3. **GPU**: It stands for graphics processing unit, used for carrying out image calculation. Broadcom video core cable is added in the device that is mainly used for playing video games.
4. **USB Ports:** Two more USB ports are introduced in this new version, setting you free from the hassle of using an external USB hub when you aim to join a number of peripherals with the device.
5. **Micro USB Power Source Connector:** This connector is used for providing 5V power to the board and the board draws 170 to 200mA more power than model B.
6. **HDMI and Composite Connection:** The audio output and video composite now both reside in a 4-pole 3.5mm socket which resides near HDMI. The power connector has also been repositioned in new B+ model and lives next to HDMI. All the power and audio video composite socket are now placed on the one side of the PCB, giving it a clean and precise look.
7. **USB Hard Drive:** The USB hard drive is available on the board that is used to boot the device. It is identical to the hard drive of regular computer where windows are used to boot the hard drive of the computer.
8. **PoE:** B+ model comes with a facility of Power over Ethernet (PoE); a new feature added in this device which allows the necessary electrical current using data cables.
9. **Other Changes:** The B+ version comes with little improvement in the features and poses slightly different layout in terms of location of the components. The SD memory slot is replaced by a micro-SD memory card slot (works similar to the previous version). The status LEDs now only contain red and green color and relocated to the opposite end of the PCB.

##### Raspberry Pi 3 B+ Technical Specifications

* + - * + CPU is 64 bits with 1GB RAM (random access memory)
        + Contains Broadcom BCM2837B0 chipset
        + Comes with 1.4GHz Quad-Core ARM Cortex-A53, 4 cores
        + Consists of 40 pin headers (26 GPIOs)
        + Stereo audio and composite video are supported by 3.5mm jack connector
        + 4 USB 2.0 ports
        + Gigabit Ethernet
        + PoE (power over Ethernet) is a major feature incorporated in this device that lacks in B model
        + 2-pin reset header
        + Micro SD socket, used to enhance the memory capacity of the board
        + Micro USB power connector, used for transferring power to the device HDMI
        + CSI camera interface
        + Comes with Wi-Fi and Bluetooth facility that were not present in previous Raspberry Pi 1 and 2 versions
        + DSI connector for official screen

##### Mechanical Dimensions

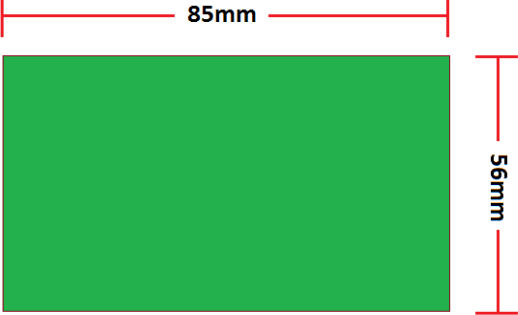
* + - * + The mechanical dimensions of this B+ are same as B version i.e., 85mm x 56mm. (Width x Height)

Fig. 3.3. Dimension of Raspberry Pi 3 B+

#### Wi-Fi Adaptor inbuilt in Raspberry Pi

* + - * An inbuilt Wi-Fi adaptor of raspberry Pi lets you connect wirelessly to networks, even if your device doesn’t have a wireless network card. Once you’ve purchased a Raspberry Pi, you'll have to see whether it connects automatically to a wireless network.
      * Inbuilt Wi-Fi has become very popular in the present time and it is now possible to get just about any add on to your computer. So, it should be no surprise to you that you can get an inbuilt Wi-Fi adapter that will enable you to connect to the Internet via a wireless router.

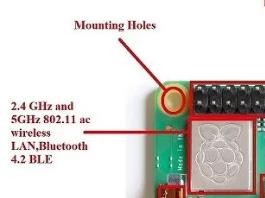


Fig. 3.4. Inbuilt Wi-Fi Adaptor

* + - * You should have a number of USB ports with your computer whether it is a desktop or a laptop. As long as you have one of these free you can plug in a Raspberry Pi 3 B+ and then this will communicate with a Wi-Fi router that will provide you with access to the Internet.

##### What does an inbuilt Wi-Fi Adapter do?

Before [wireless routers became popular,](https://www.technize.com/networking/best-routers-under-100/) you used to have a physical cable to connect your computer or laptop to an Internet router using the Ethernet system. Everything is going wireless now and with the Wi-Fi standards, it is really [easy to access the](https://www.technize.com/networking/best-wireless-access-points/) Internet via a wireless router using an inbuilt Wi-Fi adapter plugged into your desktop computer.

##### How does an inbuilt Wi-Fi Adapter work?

* + - * + If your laptop or desktop computer does not have any wireless connectivity functionality built-in then you will need to provide this if you want to communicate with a wireless router to access the Internet.
        + There are cards available that include all of the necessary electronics that you need to work with Wi-Fi standards and create a connection with a wireless router. In the case of a inbuilt Wi-Fi adapter, all of the required electronics are provided on a smaller scale inside a Raspberry Pi stick that you simply plug into a free USB port.

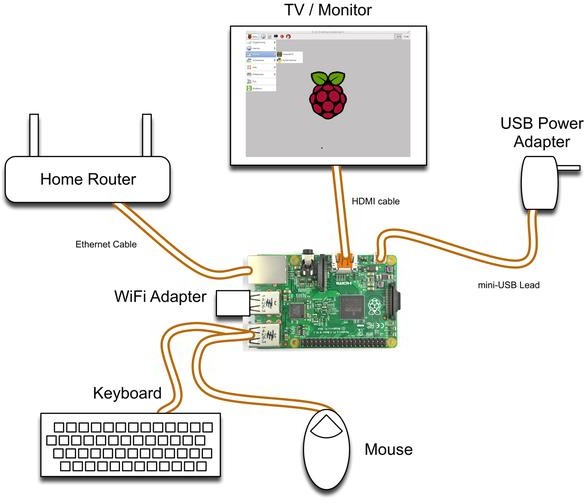


Fig. 3.5**.** Inbuilt Wi-Fi adapter working

* + - * + With inbuilt Wi-Fi adapters you will have an antenna built into the stick itself. This is used to pick up the signals from your wireless router. These tend to work reasonably well if your computer is fairly close to the wireless router.
        + You can also purchase a USB Wi-Fi adapter that includes an external antenna which helps to pick up Wi-Fi signals from your router if your computer is some distance away. These are more expensive than the USB Wi-Fi adapters that have the antenna built-in but they can be very useful if you tend to move around a lot.
        + When you plug your Raspberry Pi 3 B+ adapter into a USB port on your computer it will be able to directly communicate with your processor and other important parts of your system. It is a lot easier to install an inbuilt Wi-Fi adapter than it is to install a [PCI-E Wi-Fi adapter](https://www.technize.com/networking/best-pcie-wifi-cards/) in your computer.

#### USB Fingerprint Scanner

* + - * A fingerprint scanner is a type of technology that identifies and authenticates the fingerprints of an individual in order to grant or deny access to a computer system or a physical facility.
      * It is a type of biometric security technology that utilizes the combination of hardware and software techniques to identify the fingerprint scans of an individual.



Fig. 3.6 R307 Fingerprint Scanner

* + - * A fingerprint scanner typically works by first recording fingerprint scans of all authorized individuals for a particular system or facility. These scans are saved within a database. The user requiring access puts their finger on a hardware scanner, which scans and copies the input from the individual and looks for any similarity within the already-stored scans. If there is a positive match, the individual is granted access
      * Fingerprint scanners most commonly use an individual's thumbprint as identification.
      * Fingerprint scanners are security systems of biometrics. They are used in police stations, security industries, smartphones, and other mobile devices.

##### How does a Fingerprint Optical Scanner work?

* + - * + A fingerprint scanner system has two basic jobs -- it needs to get an image of your finger, and it needs to determine whether the pattern of ridges and valleys in this image matches the pattern of ridges and valleys in pre-scanned images.
        + Only specific characteristics, which are unique to every fingerprint, are filtered and saved as an encrypted biometric key or mathematical representation. No image of a fingerprint is ever saved, only a series of numbers (a binary code), which is used for verification. The algorithm cannot be reconverted to an image, so no one can duplicate your fingerprints.

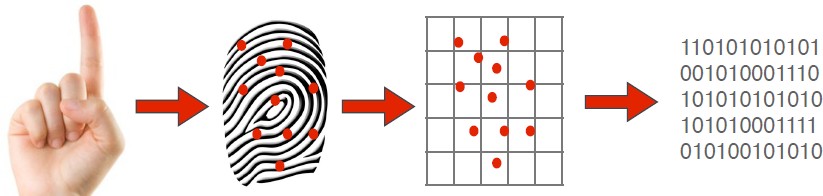


Fig. 3.7 Working of Fingerprint Scanner

##### Advantages of Fingerprint Authentication

* + - * + It is highly accurate.
        + It is unique and can never be the same for two persons.
        + It is the most economical technique.
        + It is easy to use.
        + Use of small storage space.

#### Jumper Wire

Jumper wires are simple wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used [with breadboards](https://blog.sparkfuneducation.com/what-is-a-breadboard) and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wires

##### Types of Jumper Wires

Jumper wires typically come in three versions:

* + - * + male-to-male,
        + male-to-female
        + female-to-female.

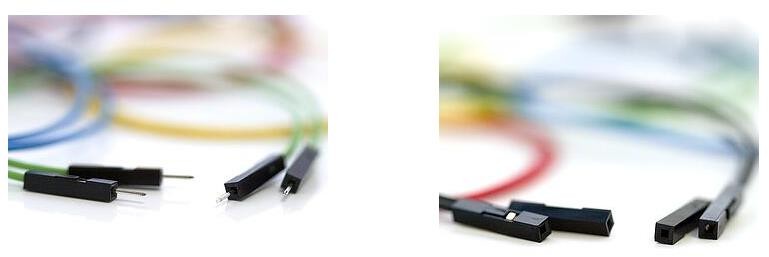


Fig. 3.8. Male to Male Jumper Wires Fig. 3.9. Female to Female Jumper Wires

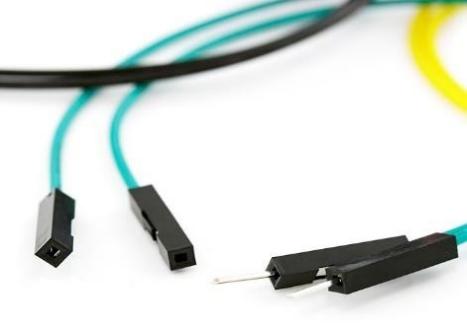


Fig. 3.10. Male to Female Jumper Wires

The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male- to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you’ll need.

* 1. **Software**

Basic overview of the software technologies used in our project discussed below:

#### Raspbian OS

Raspbian OS is one of the official Operating systems available for free to download and use. The system is based on Debian Linux and is optimized to work efficiently with the Raspberry Pi computer. As we already know an OS is a set of basic programs and utilities that runs on a specified hardware, in this case the Pi. Debian is very lightweight and makes a great choice for the Pi. The Raspbian includes tools for browsing, python programming and a GUI desktop.

##### Setting Up Raspbian OS:

Let’s first connect the board with all the necessary accessories to install and run an operating system.

**Step 1:** Take the Pi out of its anti-static cover and place it on the non-metal table. **Step 2:** Connect the display – Connect the HDMI cable to the HDMI port on the Pi and the other end of the HDMI cable to the HDMI port of the laptop.

**Step 3:** Connect your Ethernet cable from the Router to the Ethernet port on the Pi.

**Step 4:** Connect your USB mouse to one of the USB ports on the Pi.

**Step 5:** Connect your USB Keyboard to the other USB port on the Pi.

**Step 6:** Connect the micro-USB charger to the Pi but don’t connect it to the power supply yet.

**Step 7:** Flash the SD Card with the Raspbian OS.

Integrating the Raspbian OS with the Raspberry Pi using the SD card:

* + - * 1. To prepare the card for use with the Pi we will need to put an OS on the card. We certainly cannot drag and drop the OS files on to the card but the flashing the card is not too difficult either.
        2. Since we have already decided to install Raspbian, lets download the RASPBIAN image from the following link. <http://www.raspberrypi.org/downloads/>.
        3. Unzip the contents of the Zip file into a folder on your machine, one of the unzipped files would be a .img file which is what needs to be flashed on to the SD card.
        4. Flashing from Linux instructions.

Start the terminal on your Linux OS

Insert the empty SD Card into the card reader of your machine.

Type sudo fdisk -l to see all the disks listed. Find the SD card by its size, and note the device address (/dev/sdX, where X is a letter identifying the storage device. Some systems with integrated SD card readers may use

/dev/mmcblkX— format, just change the target in the following instructions accordingly).

Use cd to change to the directory with the .img file you extracted from the Zip archive.

* + - * 1. Type sudo dd if=imagefilename.img of=/dev/sdX bs=2M to write the file imagefilename.img to the SD card connected to the device address. Replace

imagefilename.img with the actual name of the file extracted from the Zip archive. This step takes a while, so be patient! During flashing, nothing will be shown on the screen until the process is fully complete.

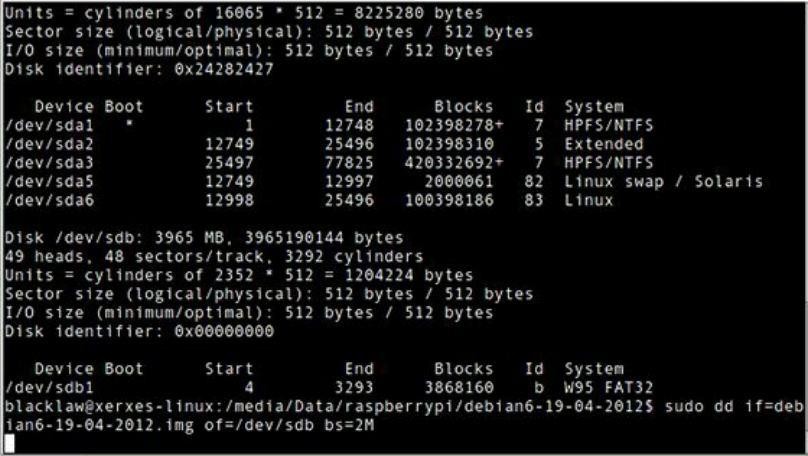


Fig. 3.10. Setting up Raspbian OS in Linux

#### Firebase Realtime Database

* The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client. When you build cross- platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.
* Store and sync data with our NoSQL cloud database. Data is synced across all clients in real time, and remains available when your app goes offline.

##### Key Capabilities

* + - * 1. **Realtime: -** Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.
        2. **Offline: -** Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.
        3. **Accessible from Client Devices: -** The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.
        4. **Scale across multiple databases: -** With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

##### How Does it work?

* The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.
* The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.
* The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to

build a great real time experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then [structure it accordingly](https://firebase.google.com/docs/database/web/structure-data).

#### Python

* + - * Python, designed by Guido van Rossum and developed by the Python Software Foundation is a renowned programming language. It was developed in 1991 and it has become one of the most popular high level programming languages. It follows simple syntax that is easily understandable even by those who are not from a programming background. It supports scalable and reliable development. The commands are written as sentence structures, allowing the user to get a brief of the idea just by a glance. In comparison with other languages python has the least redundancies.
      * It is supported by all the major operating systems such as Windows, MacOS, Linux and UNIX. There is no step for compiling, hence it makes the edit-test-debug cycle astonishingly faster than other languages. If the program doesn’t catch any exception, the interpreter prints a stack trace, making the task easy for the programmer.
      * Python offers a variety of data structures to ease our work. It also supports various libraries that can be imported using the import command. It uses white spacing for structuring the code, making it easier to read and understand. Thus, Python is the most preferred programming language for Machine Learning.
      * Python is Interpreted − The programmer does not have to compile the program before executing it. This is similar to PERL and PHP
      * Python is Interactive − Python prompt is very helpful, allowing the programmer to interact and assists while typing the code.
      * Python is Object-Oriented − It is an Object-oriented programming language that uses classes.
      * Python is a Beginner's Language − Python is a considered the best language to begin with, in the journey of programming. Its easy-to-use features allows the beginners to understand the OOPS concepts thoroughly.

##### Features:

* + - * + **Easy-to-learn −** Its structure and syntax are easy to understand.
        + **Easy-to-read −** As it has a special syntax, it is easy to go through the code at a glance.
        + **Easy-to-maintain** − Python’s code is easy to maintain, even though the updates.
        + **A broad standard library** − A variety of standard libraries are available that allows the programmer to include them in the code.
        + **Interactive Mode** − Python prompt is very helpful, allowing the programmer to interact and assists while typing the code.
        + **Portable** − Python is supported by almost every platform.
        + **Extendable** − Python is scalable and modules of all the levels can be included in the code which allows the user to code efficiently.
        + **Databases** − Python has various sets of databases for a variety of interfaces.
        + **GUI Programming** − Python also supports GUI application that are supported in various operating systems such as Windows, Mac, Linux. Hence, the same code can work for almost all the OS.
        + **Scalable** − As mentioned earlier, Python is scalable and allows the addition of new modules.

# Chapter 4

### IMPLEMENTATION

The section below discusses the implementation of all the hardware and software components of the project.

* 1. **System Architecture:**

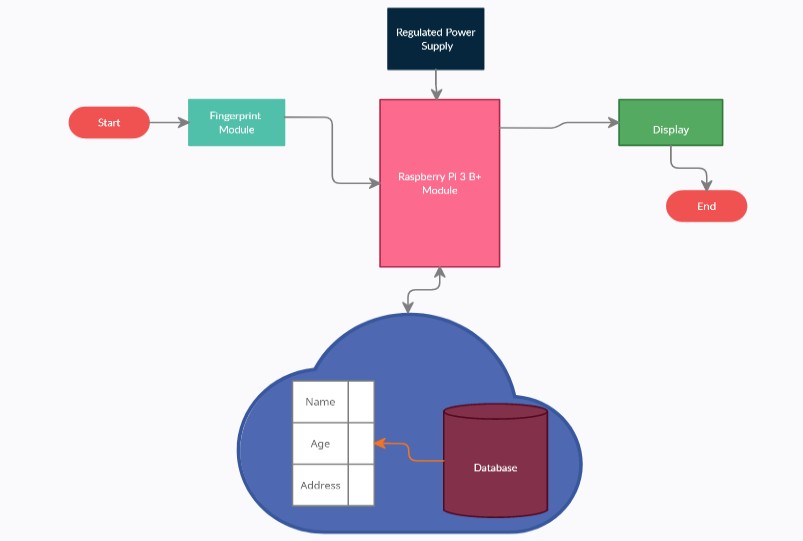


Fig. 4.1. System Architecture

#### Fingerprint Module:



Fig. 4.2. Fingerprint Sensor R307

##### The Fingerprint Sensor Description:

* + - * The R307 fingerprint scanner is used to capture biometric traits.
      * R307 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition.
      * Since it is an optical fingerprint scanner, and thus captures the traits as an image, which can either be directly downloaded, or be converted to raw data.
      * Supply Voltage: 3.6 – 6 V DC
      * Operating Current: 120mA max
      * Storage Capacity: 162 Templates
      * Interface: TTL Series
      * Baud Rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)

##### Connection of Fingerprint Sensor

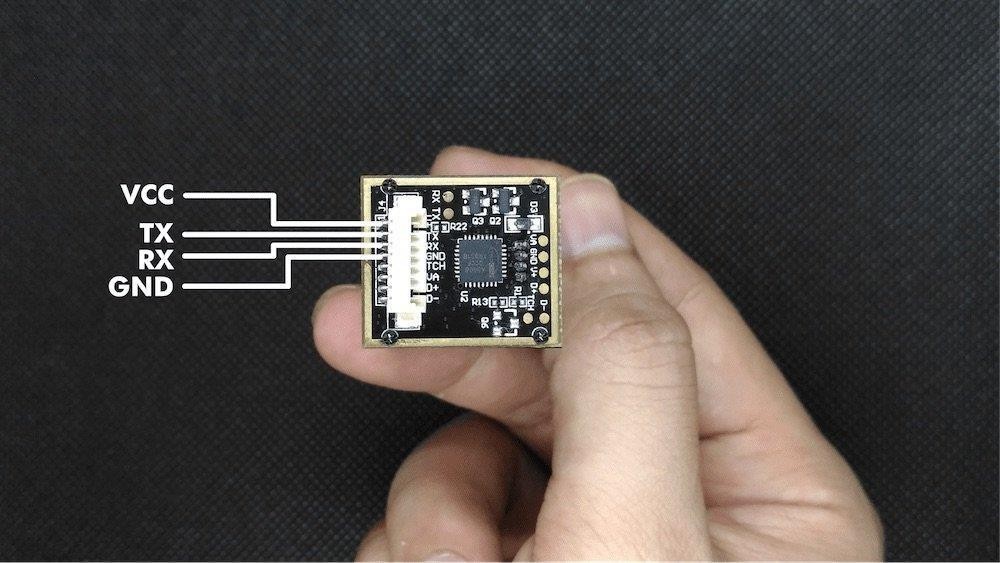


Fig. 4.3. Serial connection to the back of fingerprint module

##### Need of USB to Serial TTL UART Converter:

* + - * + We can reduce the hassle and time of connecting so many wires to your Raspberry Pi’s GPIO by using a USB to serial CP2102 TTL UART converter.
        + This converter enables you to directly plug your serial devices into your USB ports.
        + It is a great tool for embedded systems that require serial devices or sensors (like the fingerprint scanner).

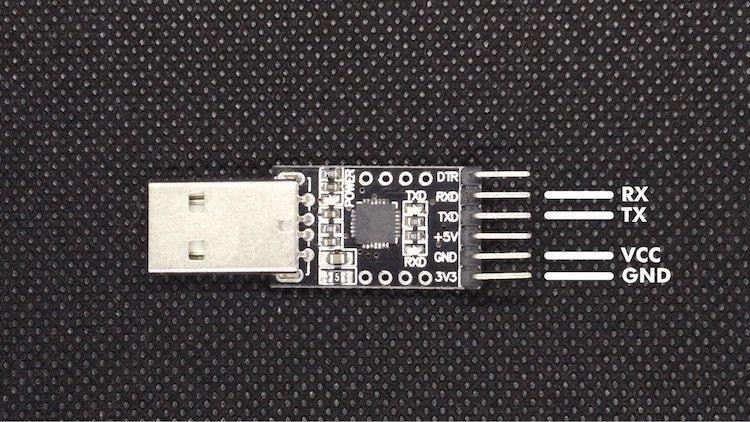
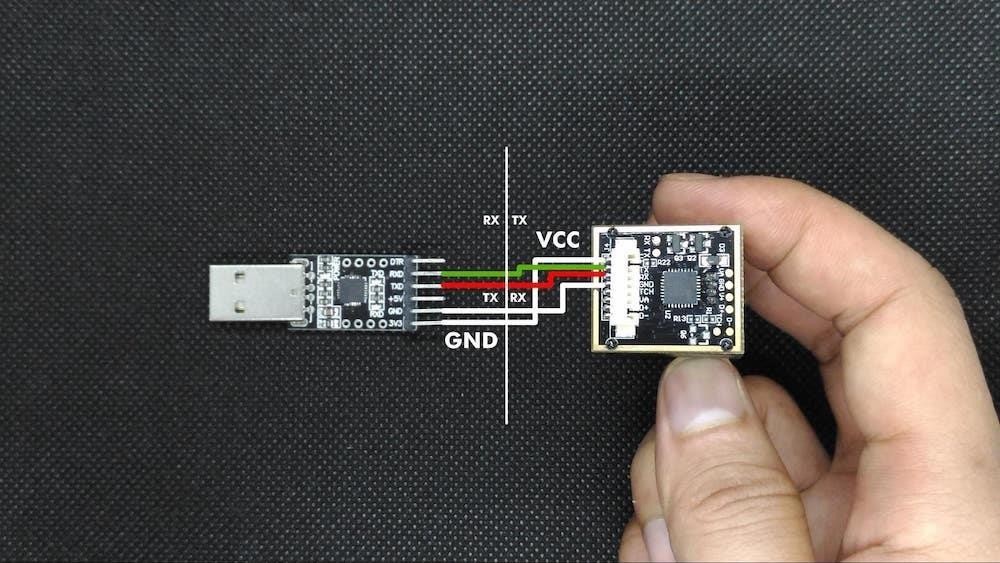


Fig. 4.4. USB to Serial TTL UART Converter

#### Raspberry pi:

##### Connecting Raspberry Pi with the Fingerprint Sensor using the converter:

* + - * By connecting the respective pins i.e., Ground (GND), Supply (VCC), Transmitter (TX) and Receiver (RX) from the serial board of the fingerprint sensor to the USB to Serial TTL UART converter using the Female-Female jumper wires.
      * This makes it possible for the Raspberry Pi to interface with the Fingerprint Sensor.



#### Database:

Fig. 4.5. Connection of the board to converter



Fig. 4.6. Firebase Realtime Database

* + - * We are using firebase as our cloud database, and it is a real-time database.
      * Here data can be pushed and pulled in real time.
      * We are storing data in key value pairs
      * Each user identified by unique identifier
      * Cloud can retrieve the user information data from database and check whether the user is valid or not for voting.

##### 4.1.3.1.Setting-up database in Google Firebase:

We visit the Firebase’s website to get started with setting-up of database. The website can be accessed using link- firebase.google.com

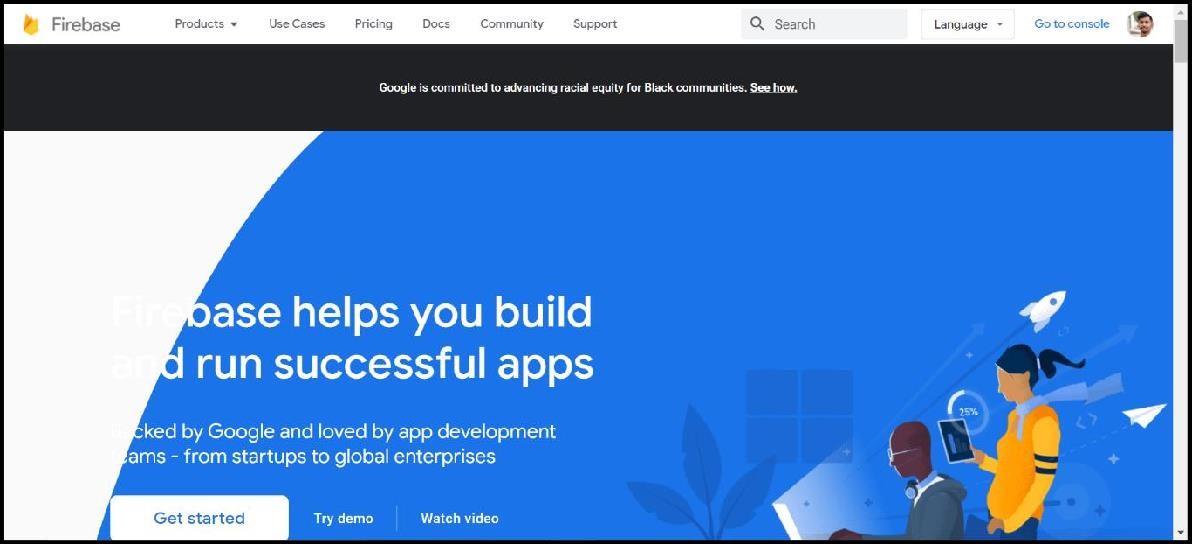


Fig. 4.7. Landing Page of Firebase

On reaching the homepage of Firebase’s website we login with our google account. Then click on console.

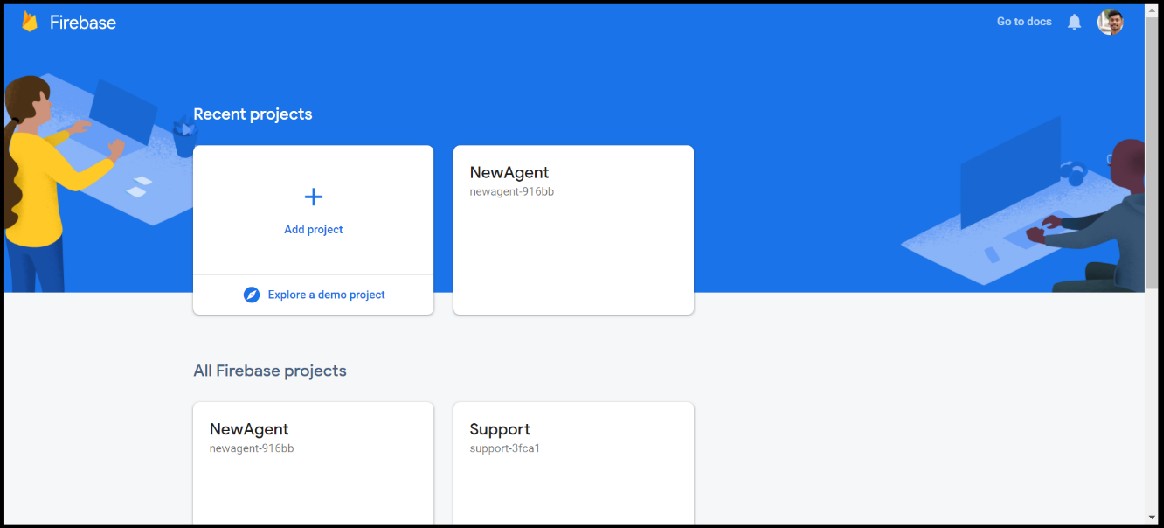


Fig. 4.8. Console page

On the console we add a new project for our database. On clicking on “Add Project”, we enter the name of the project.



Fig. 4.9. Entering the project name

Upon entering the name of the database, we give google analytics permissions and set our google analytics location as India.

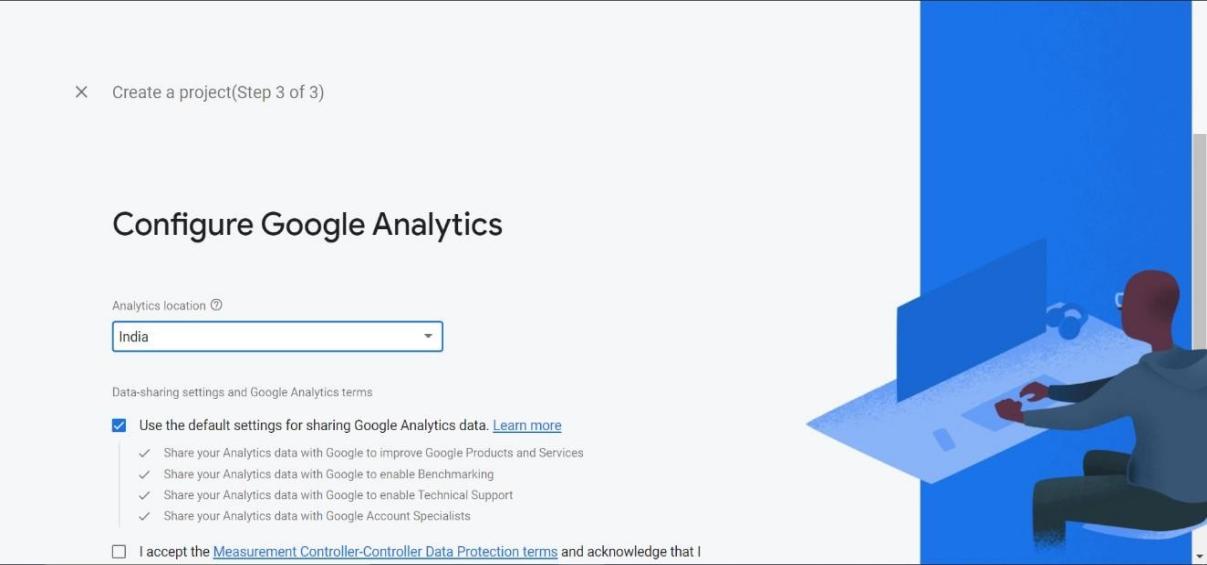


Fig. 4.10. Configuring Google Analytics

We give all the essential permissions for the analytics and then click on create project to create the project.

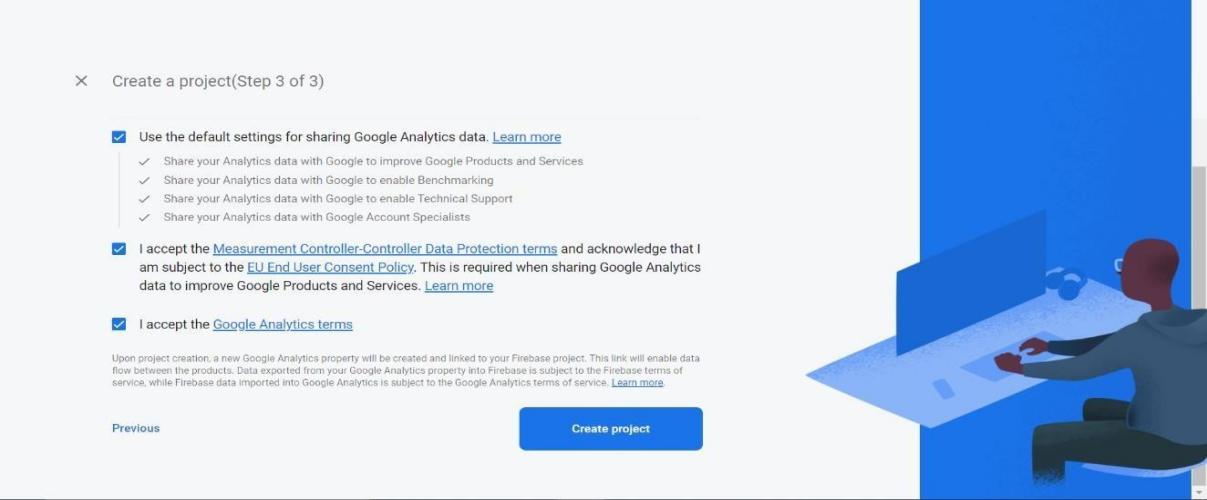


Fig. 4.11. Click on “Create Project” after all the configuring After creating the project, we see the overview page of our project.

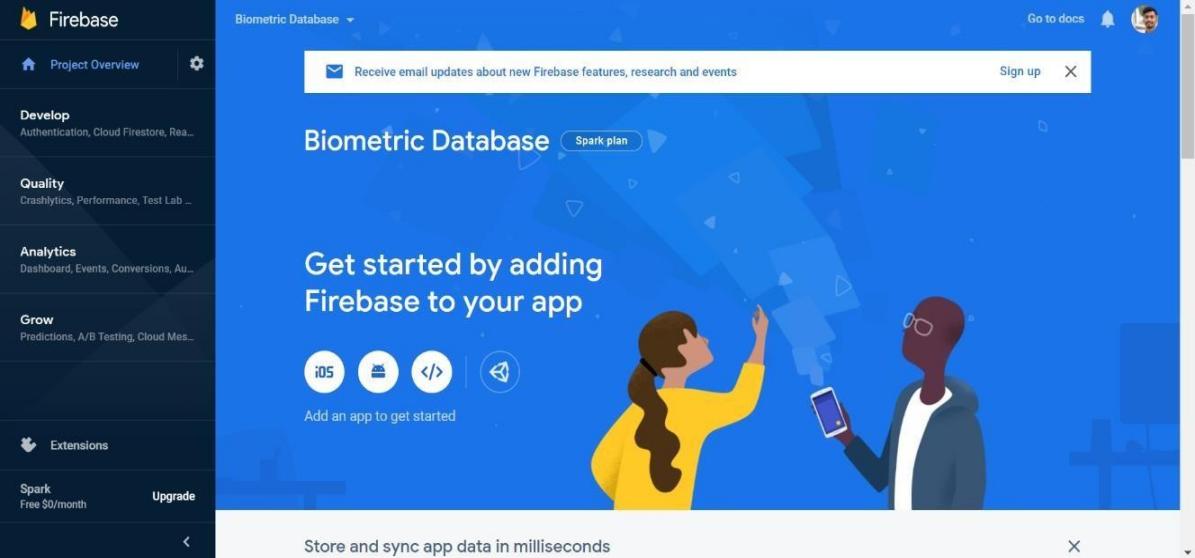


Fig. 4.12. Overview of our project

In the overview panel we go to “**Develop**”. Inside that we see “**Realtime Database**” option from where we can set-up our database.

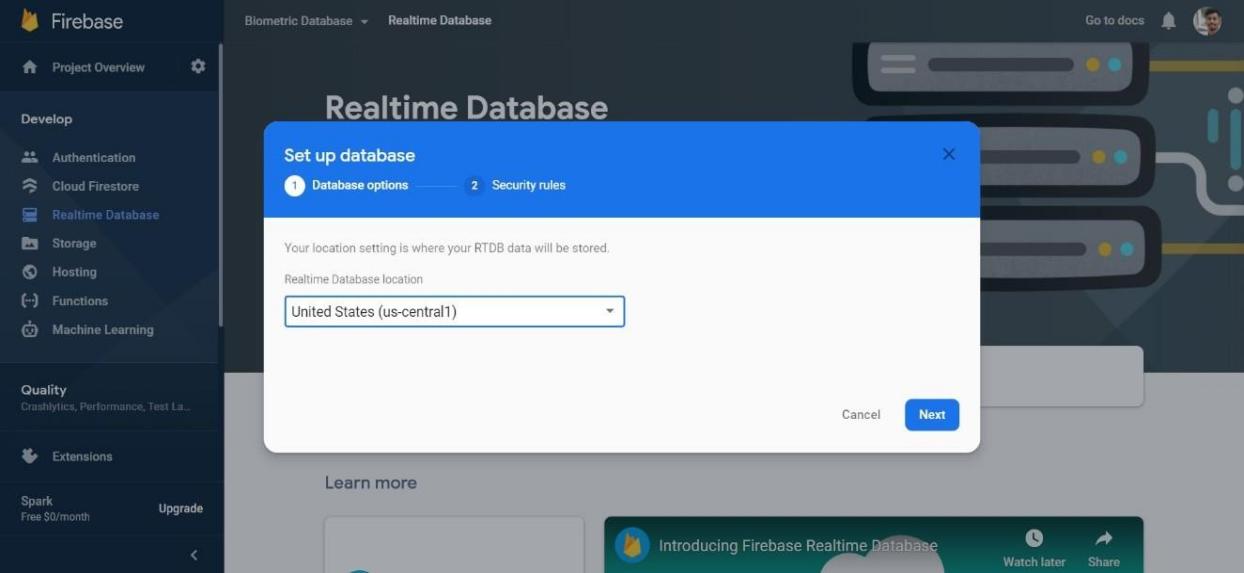
We set the database location as and (us-central1) and then select the mode of the database.

Fig. 4.13. Setting database location

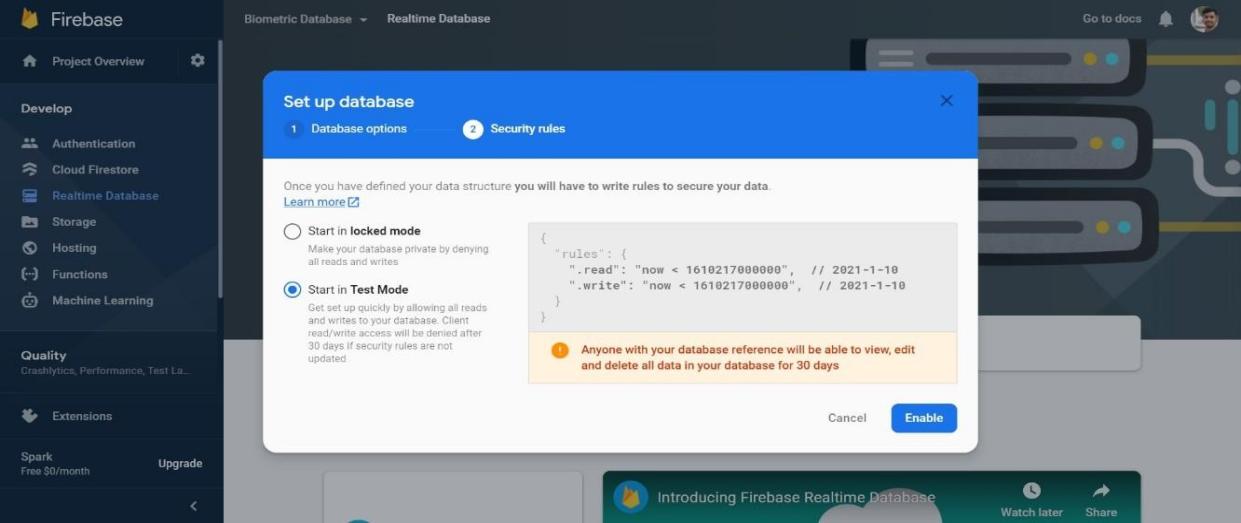


Fig. 4.14. Selecting mode of databases

After setting up all the setting we can see view of our database and its details. The database currently is **null.**

* 1. **Hardware Implementation:**

#### Input Biometric Traits:

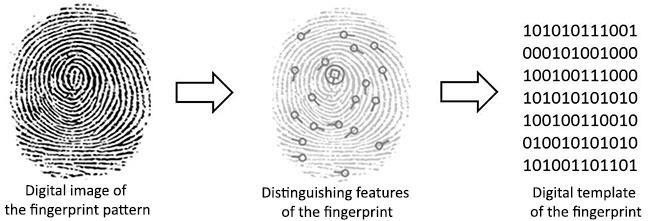


Fig. 4.15. Working of Fingerprint Sensor

* + - * Here user can place finger on fingerprint scanner,the input signal is captured as an image.
      * This needs to be converted to an array of raw data, which can be further used for feature extraction and matching.
      * This is done using the PyFingerprint.convertImage() function.

#### Raspberry pi as IOT:

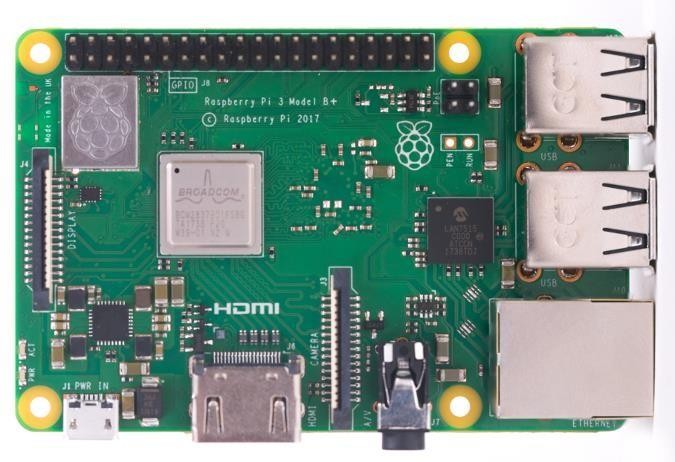


Fig. 4.16. Raspberry Pi 3 Model B+

* + - * Here we are using raspberry pi as an IOT device and we are connecting our fingerprint module.
      * The raspberry pi collects the user data and send it to the cloud for the verification and checks whether the user is enrolled or has not voted before.
      * If the above-mentioned conditions are met then it will approve a user for voting.
      * If any of the two conditions are not satisfied then user is not eligible for voting.

## Software Implementation:

#### Database:

* + - * The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code.
      * Data is persisted locally, and even while offline, real time events continue to fire, giving the end user a responsive experience.
      * The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules.
      * It defines how your data should be structured and when data can be read from or written to.
      * When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it
      * The Realtime Database API is designed to only allow operations that can be executed quickly.
      * This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness.
      * Here data can be pushed and pulled in real time.
      * Stores data in key-value pairs.
      * Each user identified by unique identifier.
      * Data can be pushed with the db.push() function.
      * Data can be retrieved with the db.get() function.
      * Data is interfaced in the form of a dictionary.

#### Python Language:

* + - * We are using python because it is an interpreted language, better for interatctive applications
      * It supports libraries for our platforms.
      * Good support for raspberry pi to communicate with our database.

#### Python Libraries:

##### pyrebase:

Well supported Python wrapper library for Firebase.

##### pyfingerprint:

Well supported Python library for multiple models of fingerprint scanners, including R307.

#### Parts of code with working:

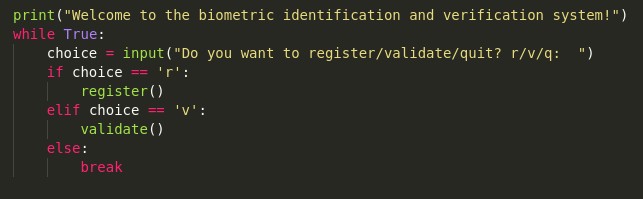


Fig. 4.17. Main Function

* + - * This function simply asks if the user wants to register or validate
      * It repeats endlessly till the user asks to quit

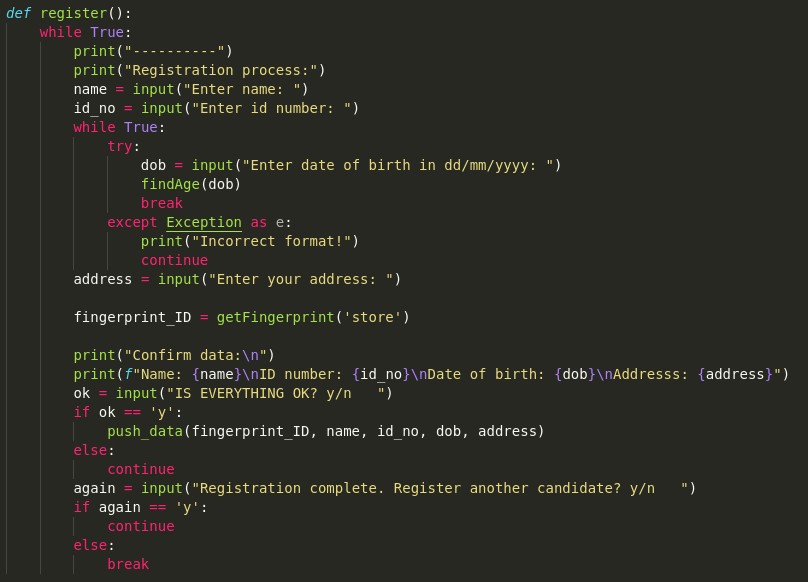


Fig. 4.18. Code for Registration

* + - * This part of the code deals with the registration of the user.
      * It asks for some details, and the date of birth, which is the main validation criteria
      * The date of birth needs to be in the correct format, and this is also accounted for.
      * It calls the getFingerprint() function with 'store' parameter, to take biometric data.
      * Once all data is acquired, user is asked to confirm, and it sends this data to the pushData() function.
      * After all this, it goes back to the main function.



Fig. 4.19. Fingerprint code to interface database and scanner

* + - * This part of the code deals with the database, as well as the driver code for the fingerprint.
      * It takes a parameter 'param' (which can be 'store' for registering or 'load' for validating), which lets it either ask the fingerprint driver code to initiate registration or validation.
      * If parameter is 'store', it simply gets the fingerprint from the driver code and returns this to the calling function.
      * If the parameter is 'load', it has to deal with an array of two objects as a result, one of which contains the parameter it's working with, and the other the data itself.
      * It works with indexed fingerprints, as well as non-indexed fingerprints, and provides appropriate data to/from the fingerprint driver code/cloud as and when required.
      * It returns an 'identity' variable which can either contain the user data or can be NULL, for registered and unregistered users respectively.

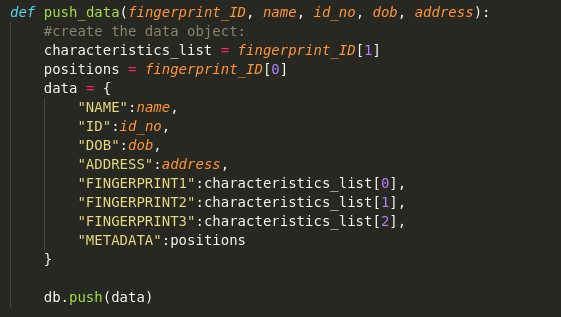


Fig. 4.20. Data Push Code

* + - * This function takes all the user details, including biometric traits, as the parameters
      * It makes a dictionary out of this
      * This dictionary is pushed to the database

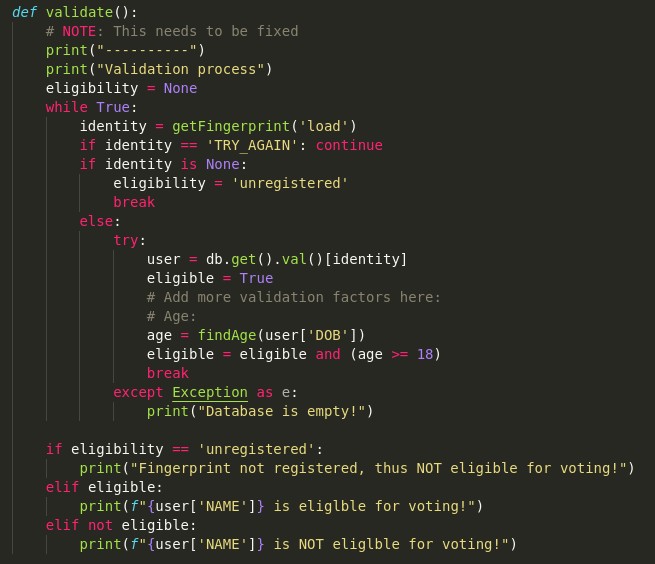


Fig. 4.21. Validation Code

* + - * This function calls the getFingerprint() function with the 'load' parameter.
      * This is stored in an identity variable, and can thus either contain user id or NULL value.
      * If it is NULL, it implies that the user is not registered.
      * If it isn't NULL, the user's details are checked and tallied with the validation criteria.
      * Depending on the result of the above, the appropriate message is displayed.



Fig. 4.22. Fingerprint Store Driver Code (part 1)

* + - * Number of samples per user is set to 3 for more accuracy.
      * Fingerprint scanner is initialized.
      * Fingerprint reader starts its operations, and the fingerprints are taken twice per run.

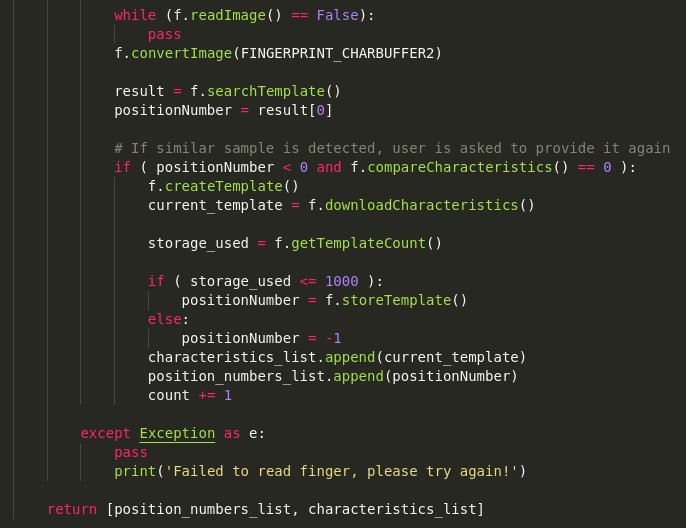


Fig. 4.23. Fingerprint Store Driver Code (part 2)

* + - * The fingerprints are converted from an image to raw data, so as to be able to extract features, work with it and upload to the cloud as text data.
      * Raw data is stored in the respective CHARBUFFERs.
      * To enable distinct fingerprints per user, we check to make sure that the features are as dissimilar to each other as possible.
      * The fingerprints are indexed, and are returned to the calling function.

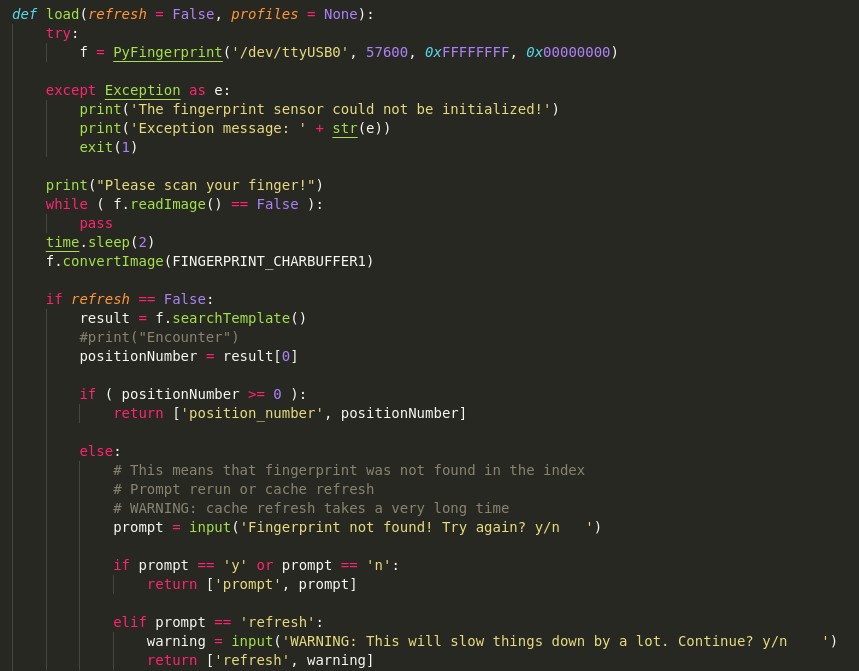


Fig. 4.24. Fingerprint load driver code (part 1)

* + - * Fingerprint scanner is initialized
      * Fingerprint reader starts its operations
      * Image is converted to raw data, and is is stored in CHARBUFFER1
      * Fingerprint is searched for in index, and if encounter happens, this is looked up in the cloud.

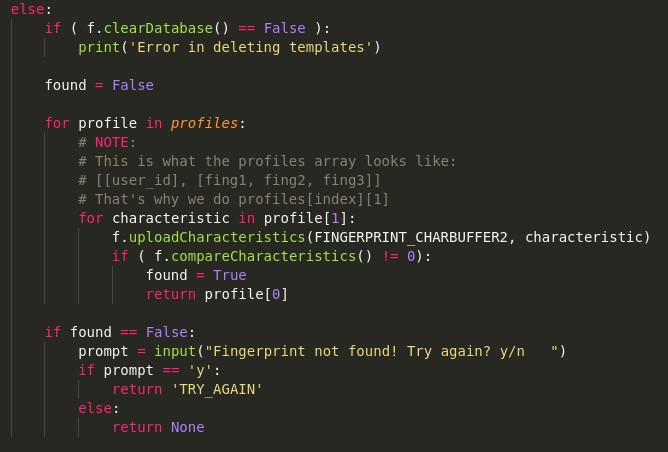


Fig. 4.25. Fingerprint load driver code part 2

* + - * If no encounter happens, it starts validation out of index.
      * Since raw fingerprint data is stored on the cloud, it needs to be retrieved from there and uploaded to CHARBUFFER2.
      * Characteristics are matched using the compareCharacteristics() function.
      * If encounter is successful, identity is returned.
      * If not, user is asked if they want to try validation again.
      * If not, it's assumed that the user is not registered.
      * Appropriate data and messages are returned and displayed respectively.

# Chapter 5

### RESULTS AND DISCUSSION



Fig 5.1. Fingerprint Sensor connected to Raspberry Pi

* Using the procedure shown in the section 4.1, we can easily connect the fingerprint sensor to the Raspberry Pi with the help of USB to Serial TTL UART converter.
* This procedure eliminates the need to configure all the Serial Board of the sensor with the GPIO pins of Raspberry Pi.



Fig 5.2. Empty database right after set-up

* This is the empty database, without any registers.
* Data is stored in key value pairs, and will start appearing once users finish registration process.

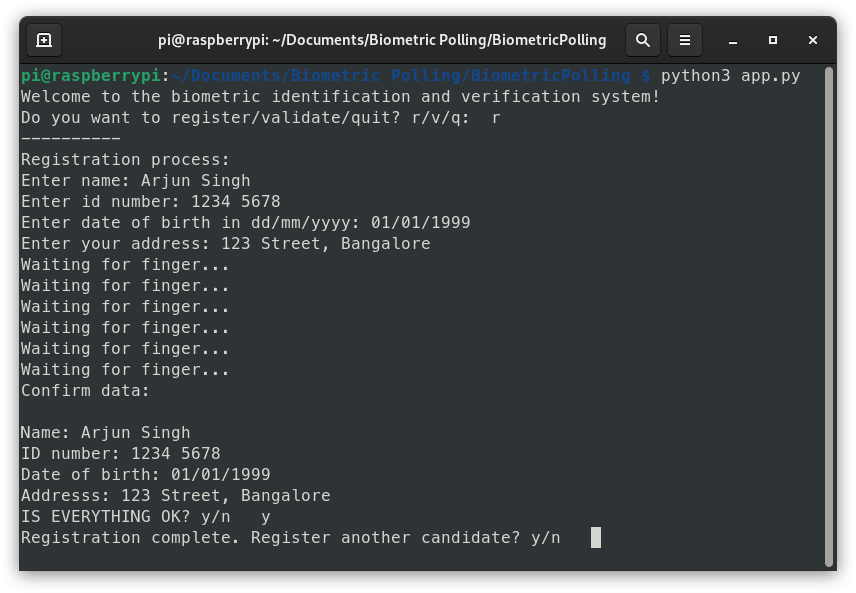


Fig 5.3. Registration process for first user

* This is the registration process for a candidate who is eligible for voting.
* Note the year of birth, which is the validation criteria. Upon entering all their details, the fingerprints are taken, and after completion of that, the user is asked to confirm the details.
* If they do confirm it, this data is pushed to the cloud.
* After every section, the users are presented with the option to either register another candidate, or to register/vote/quit.
* Thus, after pushing this data, the user is asked if they want to register another candidate or not.
* Upon entering ‘y’, the same will be repeated.



Fig 5.4. Database after pushing data for first user

* This is what the database looks like after the user has registered. As previously mentioned, data is stored in key value pairs, and a unique identifier based on time stamp is assigned to every user.

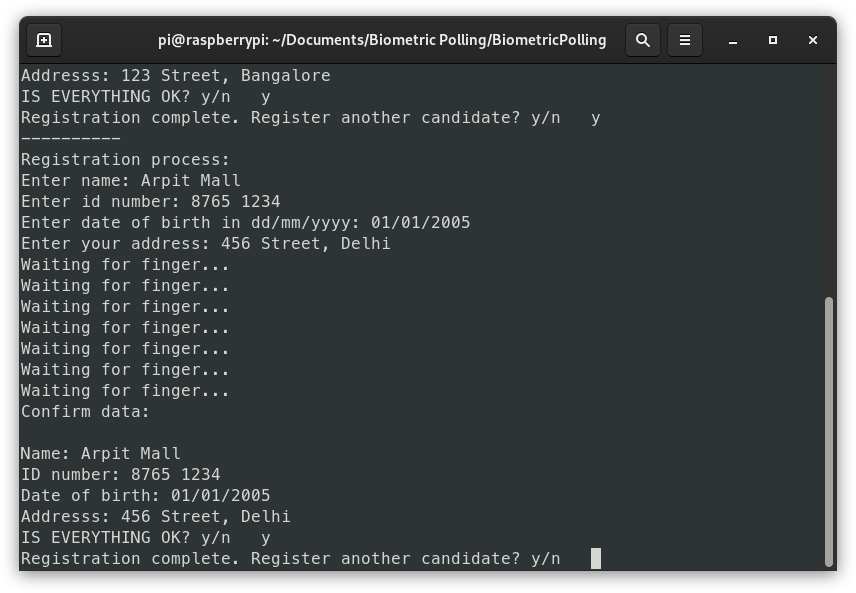


Fig 5.5. Registration process for second user

* This is the registration process for a user who is NOT eligible for voting. Note their birth year.
* The confirmation is prompted, and this data, too, is pushed to the cloud.



Fig 5.6. Database after pushing data for second user

* This is what the database looks like, after the second candidate has registered. Note the unique identifier.

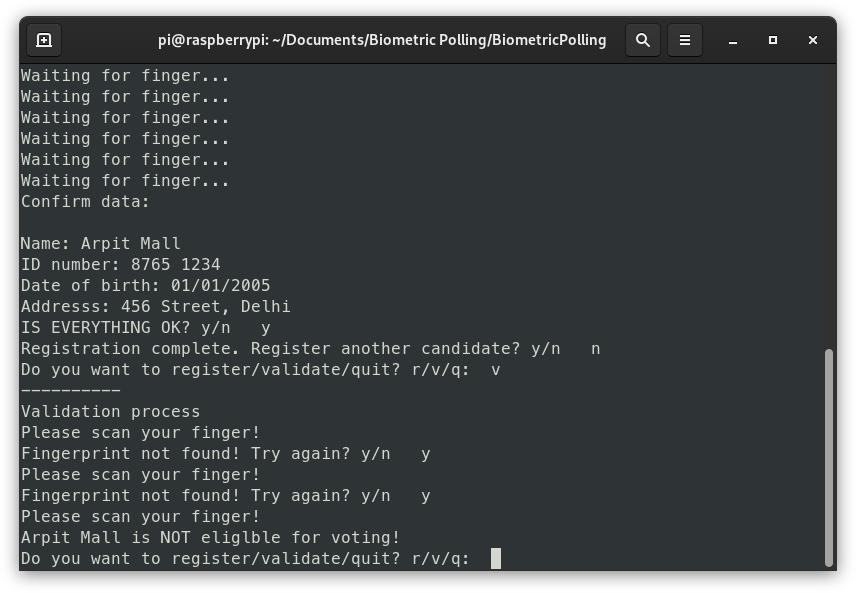


Fig. 5.7. Validation Process

* Now, we can begin the validation process. Candidate Arpit inputs their fingerprint, and this is supplied to the code. Upon calculation, it’s decided that this candidate is NOT eligible for voting.

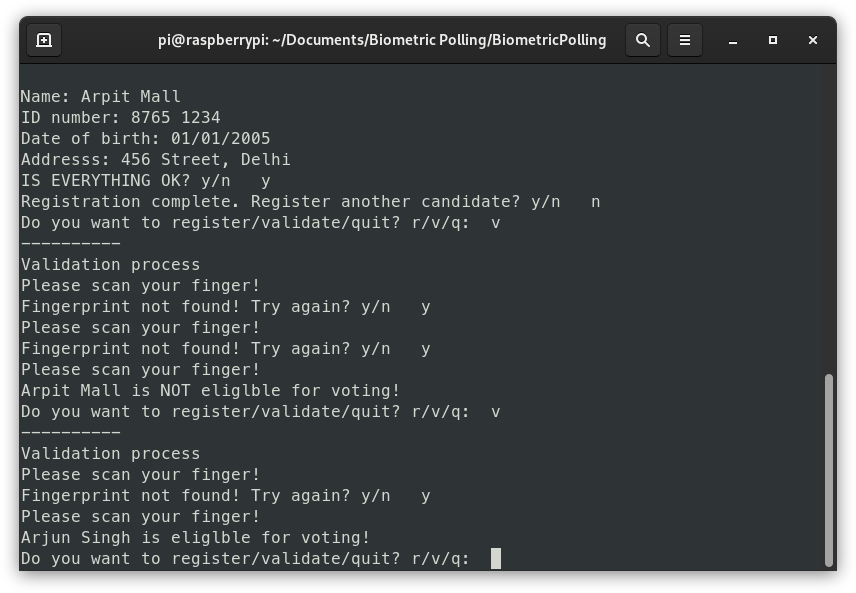


Fig. 5.9. Final result for validation

* Likewise, candidate Arjun inputs their fingerprint, and upon calculation, it is decided that this candidate meets the criteria and is thus eligible for voting.

### CONCLUSION AND FUTURE SCOPE

* The practice of voting is going on from generations. For all the important decisions for masses, a consensus is the best way to get the outcome.
* Initially, the voting was done by manually asking each individual and counting his vote. This method was not feasible for a large number of people and hence the first major phase of change happened when it was replaced by the use of ballot paper.
* Later, with the dawn of an electronic era, EVMs superseded ballot papers and now the majority of the nations all around the world use EVMs for their elections.
* But modern technologies and growth paved the way for the era of digitization and automation. This era has already affected other fields and it is the time when it should also change the way the elections work in our country.
* The impact of this project when considered along with its scale is extensive. It solves the root causes of mismanagement and potential problems that we have to deal with in a traditional approach. It eradicates the major issues such as low efficiency, time consumption, unnecessary human efforts, high amount of paperwork and restrictions regarding the polling location.
* All these played a major part in slowing down the overall process of voting in the country. These issues are faced during every election but are neglected due to lack of a better alternative.
* However, with the help of IoT and advanced high-speed biometric scanners in the project, these issues can be fixed. The project solves these major issues and also supplements the complete system by additional benefits like ease of use and maintenance.
* With the advancement in cloud storages, there is no need for local storage devices as the database can be accessed irrespective of the location and time.
* When considering all these factors that are being considered in our model, the change in the speed and reliability of the process can be increased drastically.
* The model is not only fast and efficient but also extremely cost effective.
* In this project, we address only the issues with the identification procedure of the polling. Cloud storage is used in the model which provides ease and scalability but since election is an event of national interest and trust, security of the database is also very important. For this in future, the Blockchain can be used for storing the biometrics which will make the complete system more transparent and tamper-proof.
* Blockchain can also play an important role in vote casting part. There have been several concerns regarding the security of EVMs and possibility of tampering of votes. With Blockchain based tools replacing EVMs in future, these issues can also be resolved.
* Thus, with the advancement in modern technologies, the concerns and challenges in the conduction of elections can be overcome.

# REFERENCES

1. Anubhav Mishra, Anuroop Mishra, Abhyudya Bajpai and Abhinav Mishra. (2020). Implementation of Blockchain for Fair Polling System.10.1109/ICOSEC49089.2020.9215354.
2. Dhvani Shah & Vinayak Haradi. (2016). IoT Based Biometrics Implementation on Raspberry Pi. 10.1016/j.procs.2016.03.043
3. Xue, Qingshui & Ju, Xingzhong & Zhu, Haozhi & Zhu, Haojin & Li, Fengying & Zheng, Xiangwei. (2019). A Biometric-Based IoT Device Identity Authentication Scheme. 10.1007/978-3-030-22971-9\_12.
4. Obaidat, Mohammad & Rana, Soumya & Maitra, Tanmoy & Giri, Debasis & Dutta, Subrata. (2018). Biometric Security and Internet of Things (IoT). 10.1007/978-3-319- 98734-7\_19.
5. D, Ashok Kumar & T, Ummal Sariba Begum. (2011). A Novel design of Electronic Voting System Using Fingerprint. 1. 2045-8711.
6. Srikrishnaswetha, Kone & Kumar, Sandeep & Ghai, Deepika. (2020). Secured Electronic Voting Machine Using Biometric Technique with Unique Identity Number and IOT. 10.1007/978-981-15-3172-9\_31.
7. Biometrics in 2020. Available at: [https://www.thalesgroup.com/en/markets/digital- identity-and- security/government/inspired/biometrics](https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/biometrics)
8. Internet of Things (2020). Available at: [https://www.wired.co.uk/article/internet-of- things-what-is- explained-iot](https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot)

import pyrebase import datetime import serverinfo import fingerprint

**APPENDIX - I SOURCE CODE**

firebaseConfig = serverinfo.ourServer

firebase = pyrebase.initialize\_app(firebaseConfig)

db = firebase.database();

def findAge(born):

# age = age.split['/']

#date is in form of dd/mm/yyyy

#datetime.date takes input in form of yyyy, dd, mm born = datetime.datetime.strptime(born, '%d/%m/%Y') today = datetime.date.today()

return today.year - born.year - ((today.month, today.day) < (born.month, born.day))

def getFingerprint(param): if param == 'store':

result = fingerprint.store() return result

elif param == 'load': id = db.get()

while True:

result = fingerprint.load() enum = result[0]

data = result[1]

if enum == 'position\_number': for user in id.each():

positions = user.val()['METADATA'] if data in positions:

identity = user.key() return identity

break

elif enum == 'prompt': if data == 'y':

continue else:

return None

elif enum == 'refresh': if data == 'n':

return None elif data == 'y': profiles = []

for user in id.each(): profiles.append([

user.key(), [

user.val()['FINGERPRINT1'], user.val()['FINGERPRINT2']

,user.val()['FINGERPRINT3']

]

])

identity = fingerprint.load(True, profiles) return identity

def push\_data(fingerprint\_ID, name, id\_no, dob, address): #create the data object:

characteristics\_list = fingerprint\_ID[1] positions = fingerprint\_ID[0]

data = {

"NAME":name,

"ID":id\_no,

"DOB":dob, "ADDRESS":address,

"FINGERPRINT1":characteristics\_list[0], "FINGERPRINT2":characteristics\_list[1], "FINGERPRINT3":characteristics\_list[2], "METADATA":positions

}

db.push(data)

def register(): while True:

print(" ")

print("Registration process:") name = input("Enter name: ") id\_no = input("Enter id number: ") while True:

try:

dob = input("Enter date of birth in dd/mm/yyyy: ") findAge(dob)

break

except Exception as e: print("Incorrect format!") continue

address = input("Enter your address: ") fingerprint\_ID = getFingerprint('store') print("Confirm data:\n")

print(f"Name: {name}\nID number: {id\_no}\nDate of birth: {dob}\nAddresss:

{address}")

ok = input("IS EVERYTHING OK? y/n ") if ok == 'y':

push\_data(fingerprint\_ID, name, id\_no, dob, address) else:

continue

again = input("Registration complete. Register another candidate? y/n ") if again == 'y':

continue

else:

break

def validate():

# NOTE: This needs to be fixed print(" ")

print("Validation process") eligibility = None

while True:

identity = getFingerprint('load')

if identity == 'TRY\_AGAIN': continue if identity is None:

eligibility = 'unregistered' break

else:

try:

user = db.get().val()[identity] eligible = True

# Add more validation factors here: # Age:

age = findAge(user['DOB']) eligible = eligible and (age >= 18) break

except Exception as e: print("Database is empty!")

if eligibility == 'unregistered':

print("Fingerprint not registered, thus NOT eligible for voting!") elif eligible:

print(f"{user['NAME']} is eliglble for voting!") elif not eligible:

print(f"{user['NAME']} is NOT eliglble for voting!")

print("Welcome to the biometric identification and verification system!") while True:

choice = input("Do you want to register/validate/quit? r/v/q: ") if choice == 'r':

register()

elif choice == 'v': validate()

else:

break

#### Fingerprint Module Code

import time import hashlib

from pyfingerprint.pyfingerprint import PyFingerprint

# Declare some constants here: FINGERPRINT\_CHARBUFFER1 = 0x01 FINGERPRINT\_CHARBUFFER2 = 0x02

no\_of\_samples = 3 # number of fingerprint samples per individual

# NOTE: Increasing the sample size will make the entire operation a lot slower, since it'll increase it for every individual.

def store(): try:

f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

except Exception as e:

print('The fingerprint sensor could not be initialized!') print('Exception message: ' + str(e))

exit(1)

count = 0 characteristics\_list = [] position\_numbers\_list = []

while count < no\_of\_samples: try:

print('Waiting for finger...')

while (f.readImage() == False): pass

# Convert image to characteristics and store in char buffer 1 f.convertImage(FINGERPRINT\_CHARBUFFER1)

# This is to encourage distinct samples per individual while (f.readImage() == False):

pass f.convertImage(FINGERPRINT\_CHARBUFFER2)

result = f.searchTemplate() positionNumber = result[0]

# If similar sample is detected, user is asked to provide it again if ( positionNumber < 0 and f.compareCharacteristics() == 0 ):

f.createTemplate()

current\_template = f.downloadCharacteristics() storage\_used = f.getTemplateCount()

if ( storage\_used <= 1000 ): positionNumber = f.storeTemplate()

else:

positionNumber = -1 characteristics\_list.append(current\_template)

position\_numbers\_list.append(positionNumber) count += 1

except Exception as e: pass

print('Failed to read finger, please try again!')

return [position\_numbers\_list, characteristics\_list]

def load(refresh = False, profiles = None): try:

f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

except Exception as e:

print('The fingerprint sensor could not be initialized!') print('Exception message: ' + str(e))

exit(1)

print("Please scan your finger!") while ( f.readImage() == False ):

pass time.sleep(2)

f.convertImage(FINGERPRINT\_CHARBUFFER1)

if refresh == False:

result = f.searchTemplate() #print("Encounter")

positionNumber = result[0]

if ( positionNumber >= 0 ):

return ['position\_number', positionNumber]

else:

# This means that fingerprint was not found in the index # Prompt rerun or cache refresh

# WARNING: cache refresh takes a very long time prompt = input('Fingerprint not found! Try again? y/n ')

if prompt == 'y' or prompt == 'n': return ['prompt', prompt]

')

else:

elif prompt == 'refresh':

warning = input('WARNING: This will slow things down by a lot. Continue? y/n

return ['refresh', warning]

if ( f.clearDatabase() == False ): print('Error in deleting templates')

found = False

for profile in profiles: # NOTE:

# This is what the profiles array looks like:

# [[user\_id], [fing1, fing2, fing3]]

# That's why we do profiles[index][1] for characteristic in profile[1]:

f.uploadCharacteristics(FINGERPRINT\_CHARBUFFER2, characteristic) if ( f.compareCharacteristics() != 0):

found = True return profile[0]

if found == False:

prompt = input("Fingerprint not found! Try again? y/n ") if prompt == 'y':

return 'TRY\_AGAIN' else:

return None

**INFORMATION REGARDING STUDENTS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student name | Email id | Permanent Address | Phone Number | Placement Details | Photograph | | |
| Shaik Mohammed Junaid (17BTREC007) | [shaikjunaid129@gmail.com](mailto:shaikjunaid129@gmail.com) | 11/471, MR  Street, Kadapa, YSR District, Andhra Pradesh 516001 | +91 9618743361 | Placed |  | | |
| Anuroop Mishra (17BTREC019) | [anuroopmishra21@gmail.com](mailto:anuroopmishra21@gmail.com) | 58 MIG, W-  Block Keshav Nagar, Kanpur, U.P.  208014 | +91 7275411381 | Placed |  |  |  |
| Arjun Suresh Singh (17BTREC020) | [arjun.s.singh@outlook.com](mailto:arjun.s.singh@outlook.com) | E4-1401, SNN  Raj Serenity, Begur Koppa Main Road, Yelenahalli, Bangalore  560068 | +91 9036365444 | Placed |  | | |
| Arpit Mall (17BTREC021) | [arpitmall](mailto:arpitmall02@gmail.com)[02@gmail.com](mailto:02@gmail.com) | House no. 108, Gali no. 1-A, Qutub Vihar Phase-1, New Delhi  110071 | +91 9599447656 | Not Placed |  | | |

**BATCH PHOTOGRAPH ALONG WITH GUIDE**



Dr. R. Sukumar, Incharge HOD of ECE, Professor





Shaik Mohammed Junaid 17BTREC007

Anuroop Mishra 17BTREC019

Arjun Suresh Singh 17BTREC020

Arpit Mall 17BTREC021