**IRIS RECOGNITION MODERN VOTING SYSTEM**

## A PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

**in**

## COMPUTER SCIENCE AND ENGINEERING

# ANNA UNIVERSITY :: CHENNAI – 600 025

**IRIS RECOGNITION MODERN VOTING SYSTEM**

## A PROJECT REPORT

***Submitted by***

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

**in**

## COMPUTER SCIENCE AND ENGINEERING

# 

# ANNA UNIVERSITY :: CHENNAI – 600 025



## MAY 2024

**ANNA UNIVERSITY : CHENNAI 600 025**

# BONAFIDE CERTIFICATE

Certified that this project report “**IRIS RECOGNITION MODERN VOTING SYSTEM”** is the Bonafide of **“ ”** who carried out the

project under the supervision.

## SIGNATURE SIGNATURE

## HEAD OF THE DEPARTMENT SUPERVISOR ASSISTANT PROFESSOR

Department of Computer Science and Department of Computer Science and Engineering Engineering

Trichy – 620007 Trichy – 620007

Submitted for the viva-voce held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# ACKNOWLEDGMENT

First of all we thank God for this shower of blessing and his divine help which enables us to complete the project successfully.

We extend our sincere thanks to **XXXXX, B.E., M.Sc., (Engg.)** Founder & Chairman of XXX Engineering College, Trichy for offering the means of attaining our most cherished Goal Environment.

We extend our deepest gratitude to Principal **XXXX,M.E, Ph.D.** M.I.E.T. Engineering College, Trichy, for giving us permission to do the project work successfully.

We are grateful to express our profound thanks to the Head of the department,CSE

**XXXX, ME.,** who has been the source of encouragement and moral strength throughout ourstudy period.

It gives immense pleasure to extend my sincere and heartfelt gratitude to our project Guide **XXXX M.E,** Assistant Professor for her untiring valuable and timely suggestions in dispensable situation during the period of study.

We are extremely thankful to our parents for enlightening us by providing Professional education and for their prayerful support that makes us to complete.

Also heartfelt thanks to our friends, Teaching and Non-teaching staff members who helped us to finish the project successfully.

# ABSTRACT

The use of online voting systems has been increasing in recent years as a way to increase accessibility and convenience for voters. However, ensuring the security and accuracy of these systems remains a critical challenge. This paper proposes a face and iris detection in email OTP based voting system using Convolutional Neural Network (CNN) methodology as a solution to these challenges. The system involves capturing a voter's face and iris verifying their identity using facial recognition and email OTP verification. The facial recognition process is performed using a CNN model that is trained on a database of registered voters' faces and iris. Once the voter's identity has been verified, they are able to cast their vote through the system. The combination of facial recognition and email OTP verification provides an additional layer of security against potential fraud. Additionally, the use of a secure database and a user- friendly interface ensures the privacy and accuracy of the voting process. Development and implementation of such a system must take into consideration the legal and ethical implications of using facial recognition technology in a voting system, including privacy concerns. Face and iris detection email OTP based voting system using CNN methodology offers a secure and efficient solution for online voting, while still maintaining the privacy and accuracy of the voting process.It has the potential to increase accessibility and convenience for voters while ensuring the security and accuracy of the voting process.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **viii** |
| **1** | **INTRODUCTION** | **1** |
| **2** | **LITERATURE SURVEY** | **6** |
|  | 2.1 DESIGN AND FORMAL ANALYSIS  OF ELECTRONIC VOTING PROTOCOL USING AVISPA | 6 |
|  | 2.2 IRIS NET:A SURVEY ON IRIS RECOGNITION TECHNIQUE FOR  VOTING SYSTEM | 6 |
|  | 2.3 A SECURED CLOUD-BASED E-VOTING SYSTEM USING INFORMATION DISPERSAL ALGORITHM | 7 |
|  | 2.4 ONLINE E-VOTING PROTOCOL BASED ON PALMPRINT  FEATURES | 8 |

|  |  |  |
| --- | --- | --- |
|  | 2.5 SECURE IRIS:ENHANCING VOTING SYSTEM SECURITY THROUGH DEEP LEARNING BASED IRIS RECOGNITION | 9 |
| **3** | **SYSTEM ANALYSIS** | **11** |
|  | 3.1 EXISTING SYSTEM | 11 |
|  | 3.1.1 Disadvantages | 11 |
|  | 3.2 PROPOSED SYSTEM | 12 |
|  | 3.2.1 Advantages | 14 |
| **4** | **SYSTEM REQUIREMENTS** | **15** |
|  | 4.1 HARDWARE REQUIREMENTS | 15 |
|  | 4.2 SOFTWARE REQUIREMENTS | 15 |
| **5** | **SYSTEM DESIGN** | **16** |
|  | 5.1 SYSTEM ARCHITECTURE | 16 |
|  | 5.2 DATA FLOW DIAGRAM | 17 |
|  | 5.3 UML DIAGRAM | 18 |
|  | 5.3.1 Use Case Diagram | 18 |
|  | 5.3.2 Sequence Diagram | 19 |
|  | 5.3.3 Deployment Diagram | 21 |

|  |  |  |
| --- | --- | --- |
| **6** | **MODULE DESCRIPTION** | 22 |
|  | 6.1 LIST OF MODULES | 22 |
|  | 6.1.1 Admin module | 22 |
|  | 6.1.2 View voters | 23 |
|  | 6.1.3 View voting count | 23 |
|  | 6.1.4 Registration | 23 |
|  | 6.1.5 Login | 23 |
|  | 6.1.6 Face verification | 24 |
|  | 6.1.7 OTP verification | 24 |
|  | 6.1.8 Add vote | 24 |
| **7** | **SYSTEM TESTING** | **25** |
|  | 7.1 UNIT TESTING | 27 |
|  | 7.2 INTEGRATION TESTING | 28 |
|  | 7.3 ACCEPTANCE TESTING | 28 |
| **8** | **CONCLUSION AND FUTURE ENHANCEMENT** | **29** |
|  | 8.1 CONCLUSION | 29 |
|  | 8.2 FUTURE ENHANCEMENT | 29 |
|  | APPENDICES | 30 |
|  | 1. SAMPLE CODE | 30 |
|  | 2. SCREENSHOT | 45 |
|  | REFERENCES | 50 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| FIGURE NO | FIGURE NAME | PAGE NO |
| 5.1 | System Architecture | 16 |
| 5.2 | Data Flow Diagram(Level 0) | 17 |
| 5.3 | Data Flow Diagram(Level 1) | 18 |
| 5.4 | Use Case Diagram | 19 |
| 5.5 | Sequence Diagram | 20 |
| 5.6 | Deployment Diagram | 21 |
| A.2.1 | Home Page | 45 |
| A.2.2 | Voter Registration | 45 |
| A.2.3 | Voter Login | 46 |
| A.2.4 | OTP Verification | 46 |
| A.2.5 | Add Vote | 47 |
| A.2.6 | Admin Login | 47 |
| A.2.7 | Admin Page | 48 |
| A.2.8 | View Voters | 48 |
| A.2.9 | Voting Count | 49 |

## CHAPTER 1 INTRODUCTION

The accelerated pace of urbanization in the recent decades has endangered the environment and economic sustainability by raising several social, technical and economic concerns. Intelligent sensors and recent technologies working mutually are steadily becoming more pervasive and accomplish users’ desires more effectively and efficiently. Objects with internet protocol (IP) connectivity are correlated to the internet to offer better usability in day-to-day actions.

## E-VOTING SYSTEM

Voting System is the government election processes in every democratic nation. Democracy is to allow people to vote on their interest and it is right of every individual of a democratic nation. The democratic government depends on the results of the election. Today’s world is an era of internet and technology, internet and electronics are increasing day by day. It is unavoidable to upgrade the traditional voting system and protect the security of it. The main intention of this article is to develop a new idea about voting system and also ensure security of it. Many types of voting system have been used around the world. Paper ballot voting system is an old and unsecure voting system where it is possible to fling multiple votes from same person. It runs to fling the vote with a ballot paper and a stamp. Voting system is a government selection process in every democratic nation. Democracy is meant to allow people to vote freely and voting is the right of every people of a democratic nation. The democratic government depends on the results of the election. The voting system has observed many effective changes over the past few decades, right from the traditional paper ballot voting to electronic voting and now towards the online voting. Every system tries to overcome the loop holes

of the previous system. The primary goal of this paper is to understand the traditional voting system with the recently proposed voting systems. To safeguard the security of voting system, we use voter’s images as input and training a facial recognition model using this dataset. The trained model is used to recognize voters' faces and iris is ensure that they only vote once. The major goal of this system is to guarantee that elections are conducted ethically.

## 1.2 What is blockchain-based E-voting?

In the era of remote access riding on digital interventions – from transferring funds to tele-teaching and tele-health – digital interfaces in the electoral process cannot stay far behind, especially in a democracy. In a bid to address this, the Telangana government is implementing a first-of-its-kind e-voting or electronic voting solution in the country. This is ideally meant for anyone unable to reach the polling booth or prefers to cast his or her vote remotely.

The focus initially however, is not for everyone but to those who are physically challenged, elderly and ailing. That it is based on blockchain technology makes it more secure crucial in an electoral arena, where there could be a tendency to manipulate voting.

That is where the blockchain also comes handy. He says, if any fake ID is created then the ‘hash’ value (a unique number generated by the algorithm) will not tally with the details already stored in the system about the individual (voter).

## BLOCK CHAIN

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain.

A blockchain is a distributed database that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Block chains are best known for their crucial role in crypto currency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions. The innovation with a blockchain is that it guarantees the fidelity and security of a record of data and generates trust without the need for a trusted third party.

## How Does a Blockchain Work?

The goal of blockchain is to allow digital information to be recorded and distributed, but not edited. In this way, a blockchain is the foundation for immutable ledgers, or records of transactions that cannot be altered, deleted, or destroyed. This is why block chains are also known as a distributed ledger technology (DLT).

## Blockchain Decentralization

Imagine that a company owns a server farm with 10,000 computers used to maintain a database holding all of its client’s account information. This company owns a warehouse building that contains all of these computers under one roof and has full control of each of these computers and all of the information contained within them. This, however, provides a single point of failure. What happens if the electricity at that location goes out? What if its Internet connection is severed? What if it burns to the ground? What if a bad actor erases everything with a single keystroke? In any case, the data is lost or corrupted.

What a blockchain does is to allow the data held in that database to be spread out among several network nodes at various locations. This not only creates redundancy but also maintains the fidelity of the data stored therein—if somebody

tries to alter a record at one instance of the database, the other nodes would not be altered and thus would prevent a bad actor from doing so. If one user tampers with Bit coin’s record of transactions, all other nodes would cross-reference each other and easily pinpoint the node with the incorrect information. This system helps to establish an exact and transparent order of events. This way, no single node within the network can alter information held within it.

## Is Blockchain Secure?

Blockchain technology achieves decentralized security and trust in several ways. To begin with, new blocks are always stored linearly and chronologically. That is, they are always added to the “end” of the blockchain. After a block has been added to the end of the blockchain, it is extremely difficult to go back and alter the contents of the block unless a majority of the network has reached a consensus to do so. That’s because each block contains its own hash, along with the hash of the block before it, as well as the previously mentioned time stamp. Hash codes are created by a mathematical function that turns digital information into a string of numbers and letters. If that information is edited in any way, then the hash code changes as well.

## What Is Blockchain Technology?

Blockchain technology is a structure that stores transactional records, also known as the block, of the public in several databases, known as the “chain,” in a network connected through peer-to-peer nodes. Typically, this storage is referred to as a ‘digital ledger.’

Every transaction in this ledger is authorized by the digital signature of the owner, which authenticates the transaction and safeguards it from tampering. Hence, the information the digital ledger contains is highly secure.

In simpler words, the digital ledger is like a Google spreadsheet shared among numerous computers in a network, in which, the transactional records are stored based on actual purchases. The fascinating angle is that anybody can see the data, but they can’t corrupt it.

## How Does Blockchain Technology Work?

In recent years, you may have noticed many businesses around the world integrating Blockchain technology. But how exactly does Blockchain technology work? Is this a significant change or a simple addition? The advancements of Blockchain are still young and have the potential to be revolutionary in the future; so, let’s begin demystifying this technology.

Blockchain is a combination of three leading technologies:

* Cryptographic keys.
* A peer-to-peer network containing a shared ledger.
* A means of computing, to store the transactions and records of the network.

Cryptography keys consist of two keys – Private Key and Public key. These keys help in performing successful transactions between two parties. Each individual has these two keys, which they use to produce a secure digital identity reference. This secured identity is the most important aspect of Blockchain technology. In the world of crypto currency, this identity is referred to as ‘digital signature’ and is used for authorizing and controlling transactions.

## CHAPTER 2 LITERATURE SURVEY

* 1. **TITLE : DESIGN AND FORMAL ANALYSIS OF ELECTRONIC VOTING PROTOCOL USING AVISPA**

**AUTHOR :** Htet Ne Oo,Aye Moe Aung (2017)

## DESCRIPTION :

There are many important security properties that electronic voting protocols and systems need to satisfy. They are eligibility, fairness, security, verifiability, coercion resistance and receipt-freeness. Presently, most electronic voting systems fulfill only some of the required properties. Among these properties, receipt- freeness and verifiability properties contradict each other. So, this paper aims to describe a voting protocol that satisfies both properties simultaneously without effecting security. And an electronic voting system is implemented and formally analyzed by using AVISPA tool.

## TITLE : IRIS NET : A SURBEY ON IRIS RECOGNITION TECHNIQUE FOR VOTING SYSTEM

**AUTHOR :** Jane Smith. (2018)

## DESCRIPTION :

The main objective of this comparative study is to present a comprehensive review of literature on one of the biometric identification systems namely the iris recognition system. Biometric authentication has been introduced as one of the

most fundamental security technologies. As most human phenotypes are unique, physiological features such as fingerprints, iris color, facial patterns, and geometry are used as security passwords. Because of its dependability, iris receives the greatest attention in the authentication. The segmentation, border defining, feature extraction, and matching methods are analyzed in this work. Most of the used iris datasets are also presented in the paper. The purpose of this research is to investigate current iris recognition systems and describe their phases.

## TITLE : A SECURED CLOUD-BASED E-VOTING SYSTEM USING INFORMATION DISPERSAL ALGORITHM

**AUTHOR :** John Kingsley Arthur,Kofi Sarpong Adu-manu,Charles Adjetey (2020)

## DESCRIPTION :

The introduction of the electronic voting (e-voting) system has received much attention by researchers in recent years. E-voting has been of interest to stakeholders and political party leaders in most countries (that is developed and developing) practicing democracy. Academics and technocrats have delved into technical issues related to e-voting system that could foster its smooth implementation and this have encouraged it’s full acceptance in many countries. The challenge however, is how to secure and maintain a trustworthy e-voting system devoid of security breaches especially from hacking and hijacking. The challenge still remains an open area that calls for novel designs into high level security infrastructure that may enhance and improve the security of e-voting systems to gain the full trust, acceptance, and adoption by the citizenry. Ghana deploys computerized system for registration and verification during the electoral

process. During the 2016 election, the Electoral Commission of Ghana adopted the electronic transmission of results although the elections were conducted manually. In this paper, a novel secured framework for electronic voting relying on the principle of the Information Dispersal Algorithm (IDA) is proposed. In our approach, upon voting, the voters vote record is encrypted and split for distribution on several virtual cloud servers. At the end of the voting period, the split vote records are reassembled into their original state for counting to take place. The splitting of the vote records and its assembling are performed by the IDA. The paper further discusses the design and implementation of the IDA in a typical e- voting environment.

## TITLE : ONLINE E-VOTING PROTOCOL BASED ON PALMPRINT FEATURES

**AUTHOR :** Abdallah Meraoumia,Hakim Bendjenna,Mohamed Amroune,Yahia Dris. (2020)

## DESCRIPTION :

One of application which can uses internet for information exchange is the electronic voting (e-voting). Indeed, e-voting has several advantages like the accessibility for the disabled and elderly; the ease of long-distance voting; the low costs and the greater voter turnout. Thus, the challenge that oppose e-voting process are the need of securing the transmitted information which include the voters data as well as the chosen candidates. In this paper, we propose a secure crypto-biometric scheme dedicated to online e-voting system. The fuzzy commitment concept associated with the palmprint (PLP) and palm-vein (PLV) is

the core of our system. In this study, to enhance the discriminating capability of the PLP and/or PLV feature vectors, we use Gabor filter with thresholding method. After that, the information, which will be sent, is encrypted using a random key which is then embedded in a biometric feature vector using a fuzzy commitment scheme. Then, in the central election server, the embedded encryption key is extracted using a new retrieval scheme, which is then used to decrypt the transmitted information before be processed. The experimental results showed that online e-voting based crypto-biometric system has higher performances in terms of accuracies and key retrieval.

## TITLE : SECURE IRIS : ENHANCING VOTING SYSTEM SECURITY THROUGH DEEP LEARNING BASED IRIS RECOGNITION

**AUTHOR :** Jane Smith (2021)

## DESCRIPTION :

With the growing demand for security and accurate personal authentication, as well as the new dimensions in security issues confronting the world today, a dependable and secure authentication system is essential. The iris has become a popular biometric technique because it is a highly protected internal organ of the body. Furthermore, it is very hard to surgically change it without causing significant damage to the iris. The total accuracy of the iris recognition system is determined by the iris segmentation process's performance. This paper presents a deep learning technique for improving iris segmentation performance. The proposed method employs an efficient deep learning technique (SegNet), which performs joint semantic segmentation of ocular qualities (iris and pupil) with

greater accuracy in unconstrained scenarios. These difficult circumstances limit the performance and dependability of ocular segmentation structures. Segmentation begins by denoising the pristine image with a deep convolutional neural network to address these issues (DCNN). The semantic segmentation of the iris and pupil is then accomplished with the help of a densely connected fully convolutional encoder-decoder network. Finally, for feature extraction and classification, the proposed system is implemented using a pre-trained convolutional neural network (Alex Net). The iris recognition system's performance is evaluated using five public databases: IITD, iris databases CASIA-Iris-V1, CASIA-Iris-V2 device 1, CASIA-Iris-V2 device 2, and MMU iris database. The results show that the proposed system has a high accuracy rate of 94.08 percent, 84 percent, 97.31 percent, 100 percent, and 97.7 percent, respectively and has time of execution of less than or equal to 2 minutes

## CHAPTER 3 SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

In existing system a fuzzy commitment method. Double verification scheme has been implemented in existing system. Multifactor authentication scheme has been implemented. The proofs are generation upon storage of each vote that further verifies the counting process without decrypting the content. The proposed mechanism is validated by showing the experimental results in comparison of existing mechanisms. The blockchain based e-voting system is still at its early stages. The goal of this article have proposed a transparent mechanism in order to overcome the existing issues such as cost, time, delay, computational storage, key management overhead etc. the proposed framework is evaluated against existing baseline model depends on several performance criteria and metrics.

## DISADVANTAGE

* + - * Dos attacks possible
      * Blockchain thus removing

## PROPOSED SYSTEM

The proposed voting system leverages advanced technologies such as facial and iris recognition, combined with Convolutional Neural Networks (CNNs), to ensure the ethical conduct of elections. Initially, a dataset comprising voters' facial and iris images, along with their names, is utilized to train the model. Once trained, the system employs a webcam to capture live video footage of voters, detecting their faces and iris patterns in real-time. By comparing these detections with the

stored dataset, the system prevents voter fraud by allowing each voter to cast only one vote. The voting process itself is streamlined, with voters presented a list of parties to select from, and their choices recorded efficiently. Additionally, the system sends out comprehensive voting reports via email after the process concludes. Prior to voting, voters undergo identity verification through facial and iris matching against stored database values, ensuring the integrity of the process. This modernized voting system represents a significant advancement in electoral technology, prioritizing both accuracy and accessibility. By harnessing the power of CNNs and sophisticated recognition algorithms, the system enhances the security of elections while simplifying the voting experience for citizens. Through real-time detection and comparison mechanisms, the system effectively mitigates the risk of fraudulent voting, ensuring that each eligible voter can participate in the democratic process with confidence. Moreover, the integration of identity verification measures adds an extra layer of security, safeguarding the integrity of the voting process. This proposed system embodies a forward-thinking approach to election management, poised to revolutionize the way elections are conducted, while upholding the fundamental principles of fairness and transparency.

A proposed face and iris based voting system using Convolutional Neural Network (CNN) methodology would involve the following steps and components:

**Registration**: Voters would need to register their face and iris recognition the email address with the system in advance, providing a clear and recognizable photo of their face and irisa valid email address.

**Image Capture:** On the day of voting, the voter would have their face and iris captured using a high-quality camera or webcam.

**Preprocessing:** The captured image would be preprocessed to ensure that it meets the requirements of the CNN model. This may include cropping the image, resizing it to a specific size, and converting it to a format that the model can recognize.

**Face Detection:** The preprocessed image would be fed into the CNN model, which would detect and extract the face from the image.

**Face Recognition:** The CNN model would then compare the extracted face to a database of previously registered faces to determine the identity of the individual.

**Email Verification:** If the individual is recognized, an email containing an OTP would be sent to the email address associated with their account. The voter would then need to enter the OTP in the system to verify their identity.

**Voting:** If the OTP is entered correctly, the voter would then be allowed to cast their vote.

**Secure Database:** A secure database would be used to store the information about the registered voters and their voting records, to ensure the privacy and security of the voting process.

**User Interface:** A user-friendly interface would be designed to make the voting process simple and intuitive for the voters, with clear instructions and step-by-step guidance.

**System Maintenance:** The system would need to be regularly maintained to ensure its performance and accuracy, and to address any issues that may arise during the voting process.

The combination of facial recognition and email OTP verification in this proposed system would provide an added level of security and protection against potential fraud or hacking, while still providing a fast and convenient way for individuals to cast their vote.

## Advantage

* + - * Unparalleled security with unique iris patterns.
      * Swift and precise voter authentication.
      * Accessibility for diverse voter demographics.
      * Reliable data integrity through secure storage.
      * Future-proofing against evolving threats.

## CHAPTER 4 SYSTEM REQUIREMENTS

* 1. **HARDWARE REQUIREMENTS:-**
     + processor - INTEL
     + RAM - 4 GB (min)
     + Hard Disk - 20 GB

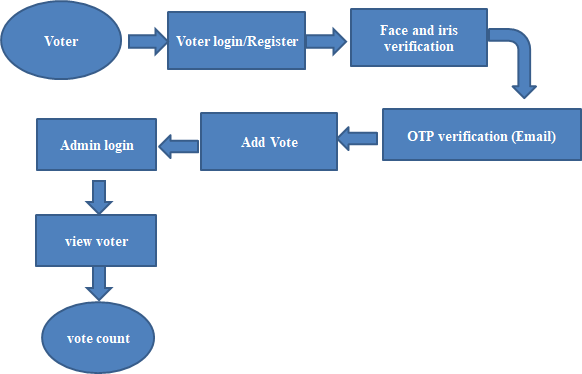
## SOFTWARE REQUIREMENTS:-

* + - Operating System : Windows 7 or 8
    - Software : Python Idle

## CHAPTER 5 SYSTEM DESIGN

* 1. **SYSTEM ARCHITECTURE**

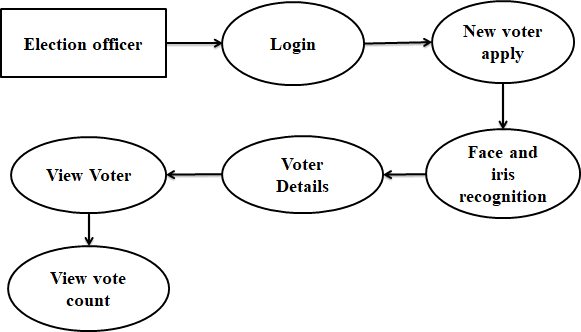
A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



## Figure 5.1:System Architecture

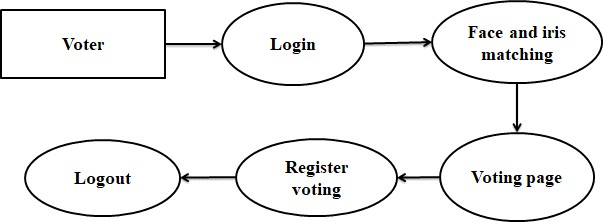
* 1. **DATA FLOW DIAGRAM:**
     + The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
     + The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
     + DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
     + DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

## DFDLevel0:



**Figure 5.2: Data Flow Diagram(Level 0)**

## DFD Level 1:

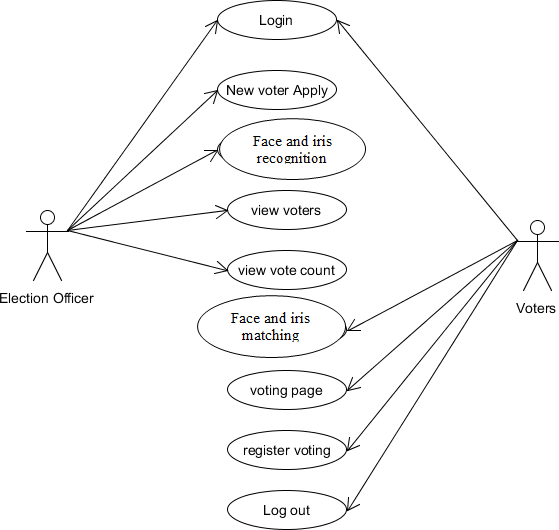


**Figure 5.3: Data Flow Diagram(Level 1)**

## UML DIAGRAMS:

* + 1. **USE CASE DIAGRAM:**

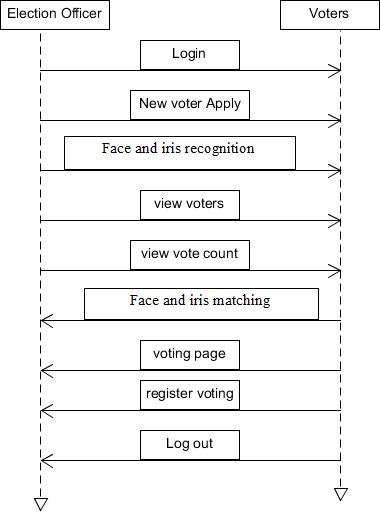
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



## Figure 5.4: Use Case Diagram

* + 1. **SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



## Figure 5.5: Sequence Diagram

* + 1. **DEPLOYMENT DIAGRAM :**

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

Database

Voters

Election Officer

## Figure 5.6: Deployment Diagram

**CHAPTER 6 MODULE IMPLEMENTATION**

## LIST OF MODULES

* + - Admin Login
    - View Voters
    - View Voting Count
    - Registration
    - Login / Logout
    - Face Verification
    - OTP Verification
    - Add Vote

## ADMIN LOGIN:

Admin can only access view the voter list and voting count.Admin\_login endpoint first checks if a POST request has been made. If so, it takes the username and password from the request.Compares it to the password stored in the admin function. If the hashes match, the admin is redirected to the admin index. If the password do not match, an "Incorrect password" message is returned. The admin\_login endpoint simply returns a welcome message.

## VIEW VOTERS

* + - * View Voters module is displayed the how many voter are vote to the election.
      * It can see the voter details as like voter id, name and email those are visible on table.
      * This module can accessed only admin.

## VIEW VOTING COUNT

* + - * This module is working for parties get how many votes in the election.
      * It can see the party name and they are get how many vote.
      * This module can accessed only admin.

## REGISTRATION

* + - * The registration module allows the user to create login Voter Id,Name and Face, iris by submitting their information like mail id.
      * By registering in the network the user can gain access to the resources stored in the server.

## LOGIN

* + - * In this module the user can login by using their unique Vote Idand Face, iris.
      * The login module verifies the user given Vote Id and Face, iris with the stored Face and iris in the server.
      * If the Face and iris is matched then the user can access the resources.
      * If it does not match, then the user does not allowed to access the resources

## FACE VERIFICATION

* + - * In this module verified to your registered face.
      * The current face is matched to the registered voter id face and iris then allowed to generate the OTPto the candidate.
      * The face and iris not matched to voter id the system is not allowed to user.

## OTP VERIFICATION

* + - * This module is extra verification to the candidates after complete the face and iris verification then OTP will generate and send to the user email.
      * Enter the correct OTP on the user only allowed to vote. If doesn’t match the OTP the system is stopped.

## ADD VOTE

* + - * This module is complete the all verifications then displayed the party names and allow to enter the vote.

# CHAPTER 7

## SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and

consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined. **System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least

its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

## Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

.you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

## Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

## Test objectives

* + - All field entries must work properly.
    - Pages must be activated from the identified link.
    - The entry screen, messages and responses must not be delayed.

## Features to be tested

* + - Verify that the entries are of the correct format
    - No duplicate entries should be allowed
    - All links should take the user to the correct page.

## INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# CHAPTER 8

**CONCLSUION AND FUTURE ENHANCEMENT**

## CONCLUSION

In conclusion, the proposed voting system represents a significant advancement in election technology, offering a comprehensive solution to the challenges faced by election commissions worldwide. By leveraging cutting-edge technologies such as face and iris recognition, along with email OTP authentication, the system ensures a secure and efficient voting process that mitigates the risk of fraudulent activities. Through the integration of convolutional neural network (CNN) methodologies, voter authentication is strengthened, enhancing the integrity and accuracy of the electoral process. Moreover, by streamlining verification procedures and reducing human intervention, this innovative system not only addresses current concerns but also sets the stage for a more reliable and inclusive voting experience in the future. With its forward- thinking approach and commitment to enhancing democratic processes, the proposed system has the potential to revolutionize the way elections are conducted, fostering trust, transparency, and participation among voters.

## FUTURE ENHANCEMENT

* + - **Enhanced security:** Iris recognition adds an extra layer of authentication, reducing the risk of fraudulent voting.
    - **Increased accuracy:** By verifying voters' identities through their unique iris patterns, the system minimizes errors in the voting process.
    - **Streamlined process**: Iris recognition technology speeds up voter verification, leading to quicker and more efficient elections.
    - **Inclusivity:** Iris recognition can benefit voters with disabilities or those who face difficulties with traditional identification methods.
    - **Data integrity:** The system maintains the integrity of voter data by securely storing and verifying iris patterns.
    - **Future-proofing:** Adopting advanced biometric technology ensures the voting system remains resilient against evolving threats and challenges.

# APPENDIX:1

## SAMPLE CODE

from flask import \* importmysql.connector import cv2

import torch importos importnumpy as np

fromtorchvision import transforms

fromfacenet\_pytorch import MTCNN,InceptionResnetV1 import time

import pickle

from PIL import Image from random import \* fromemail\_otp import \* app = Flask( name )

app.secret\_key = 'EmailAuthenticationByShivamYadav2021'

mydb = mysql.connector.connect(host="localhost",user="root",password="",database="inf o")

mycursor = mydb.cursor() @app.route("/")

def homepage(): returnrender\_template('index.html')

@app.route("/AdminLogin") defAdminLogin(): returnrender\_template('training.html') @app.route("/logi")

deflogi():

returnrender\_template('login.html') @app.route("/AtmLogin") defAtmlLogin():

return render\_template('atmlogin.html') @app.route("/adlog")

defadlog():

return render\_template("admin.html") @app.route('/val',methods=['GET', 'POST']) defval():

ifrequest.method == 'POST':

ifrequest.form.get('username') == 'abc' and request.form.get('password') == '123': returnrender\_template('admin.html')

else:

returnrender\_template('login.html', msg='Invalid Username or Password') else:

returnrender\_template('login.html') @app.route("/Admin", methods=['GET', 'POST']) def Admin():

ifrequest.method == 'POST': card = request.form['name']

un = request.form['ename'] email=request.form['email']

insertQuery = "INSERT INTO facetb VALUES ('" + card + "','" + un + "','" + email + "')"

mycursor.execute(insertQuery) mydb.commit()

current\_otp = sendEmailVerificationRequest(

receiver=email) # this function sends otp to the receiver and also returns the same otp for our session storage

session['current\_otp'] = current\_otp image\_size = 600

frame\_rate = 64

vid\_len = 20 # Length of video in seconds

#card\_no = input('Enter the card number: ').strip() #person\_name = input('Enter the person\'s name: ').strip()

device = torch.device(

"cuda" if torch.cuda.is\_available() else "cpu")

# Save all face images of a person as a pickle file defsave\_face\_images(frames, boxes):

transform = transforms.Compose([ transforms.Resize((160, 160)), transforms.ToTensor()

for f in range(len(frames)): img = np.asarray(frames[f]) box = boxes[f]

iflen(box.shape) == 3:

# Go into loop only when there is atleast 1 face in image # Loop for num of boxes in each image

for b in range(box.shape[1]):

start = (np.clip(int(box[0][b][0]) - 15, 0, 480),

np.clip(int(box[0][b][1]) - 50, 0, 640))

end = (np.clip(int(box[0][b][2]) + 15, 0, 480),

np.clip(int(box[0][b][3]) + 20, 0, 640)) crop\_pic = img[start[1]:end[1], start[0]:end[0]] img\_crop = Image.fromarray(crop\_pic) img\_crop = transform(img\_crop)

img\_crop = torch.unsqueeze(img\_crop, 0) save\_tensor = model(img\_crop) returnsave\_tensor

v\_cap = cv2.VideoCapture(0) v\_cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, image\_size) v\_cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, image\_size) count = 1

prev = 0 try:

os.mkdir(card) exceptFileExistsError:

pass

mtcnn = MTCNN(image\_size=image\_size, keep\_all=True, device=device, post\_process=True)

model = InceptionResnetV1(pretrained='vggface2', classify=False).eval()

start = time.time() frames = []

boxes = []

print( 'Try to keep your face at the centre of the screen and turn ur face slowly in order to capture diff angles of your face')

time.sleep(3)

print('A window will pop up in abt 3 seconds') time.sleep(3)

save\_tensor = None

# 20 sec loop to input truth face images while True:

time\_elapsed = time.time() - prev curr = time.time()

ifcurr - start >= vid\_len:

break

ret, frame = v\_cap.read()

cv2.imshow('Recording and saving Images', frame)

iftime\_elapsed> 1. / frame\_rate: # Collect frames every 1/frame\_rate of a second prev = time.time()

frame\_ = Image.fromarray(frame) frames.append(frame\_)

batch\_boxes, prob, landmark = mtcnn.detect(frames, landmarks=True) frames\_duplicate = frames.copy()

boxes.append(batch\_boxes) boxes\_duplicate = boxes.copy()

# show imgs with bbxs

ifsave\_tensor == None:

save\_tensor = save\_face\_images(frames\_duplicate, boxes\_duplicate) else:

temp = save\_face\_images(frames\_duplicate, boxes\_duplicate) if temp is not None:

save\_tensor = torch.cat([temp, save\_tensor], dim=0) print(save\_tensor.shape)

count += 1 frames = [] boxes = []

if cv2.waitKey(1) & 0xFF == ord('q'): break

# Open file for pickling face\_file = open(card + '/' + un, 'ab') pickle.dump(save\_tensor, face\_file) face\_file.close()

v\_cap.release() cv2.destroyAllWindows() returnrender\_template('facecam.html')

@app.route('/face', methods=['GET', 'POST']) def face():

ifrequest.method == 'POST': card\_number = request.form['card']

#un = request.form['email']

insertQuery = "INSERT INTO atmtb VALUES ('" + card\_number + "')" mycursor.execute(insertQuery)

mydb.commit()

transform = transforms.Compose([ transforms.Resize((160, 160)), transforms.ToTensor()

])

# Parameters

#card\_number = input('Enter the card\_number : ').strip() frame\_rate = 16

prev = 0

batch\_size = 32

image\_size = 600

threshold = 0.85

device = device = torch.device(

"cuda" if torch.cuda.is\_available() else "cpu")

bbx\_color = (0, 255, 255) current\_person = None defdetect\_imgs(img): globalcurrent\_person

person\_ = None img = transform(img)

img = torch.unsqueeze(img, 0) img = model(img)

minimum = torch.tensor(99)

for face\_, name in zip(faces, face\_names):

temp = torch.min(torch.norm((face\_ - img), dim=1)) if temp < minimum and temp < threshold:

minimum = temp

person\_ = name current\_person = name

return person\_, minimum.item() defshow\_images(frames, boxes, color):

temp = None

for f in range(len(frames)):

img = np.asarray(frames[f]) box = boxes[f] iflen(box.shape) == 3:

# Go into loop only when there is atleast 1 face in image # Loop for num of boxes in each image

for b in range(box.shape[1]):

start = (np.clip(int(box[0][b][0]) - 15, 0, 600),

np.clip(int(box[0][b][1]) - 20, 0, 600))

end = (np.clip(int(box[0][b][2]) + 15, 0, 600),

np.clip(int(box[0][b][3]) + 15, 0, 600))

img = cv2.rectangle(img, start, end, color, 2) crop\_pic = img[start[1]:end[1], start[0]:end[0]] crop\_pic = Image.fromarray(crop\_pic)

person, diff = detect\_imgs(crop\_pic) if person is not None:

cv2.putText(img, person + ': ' + '{:.2f}'.format(diff), (start[0], start[1] - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, color, 1)

temp = 1 else:

cv2.putText(img, 'Unknown' + ': ' + '{0}'.format(diff), (start[0], start[1] - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, color, 1)

temp = 0 cv2.imshow('Detection', img) if temp == 1:

return 1 else:

return 0

# Init MTCNN object

mtcnn = MTCNN(image\_size=image\_size, keep\_all=True, device=device, post\_process=True)

model = InceptionResnetV1(pretrained='vggface2', classify=False).eval() # Real time data from webcam

frames = [] boxes = []

# Load stored face data related to respective card number faces = []

face\_names = [] face\_file = None try:

for person in os.listdir(card\_number):

face\_file = open(card\_number + '/' + person, 'rb')

ifface\_file is not None:

face = pickle.load(face\_file) faces.append(face) face\_names.append(str(person)) exceptFileNotFoundError:

print('Face data doesnt exist for this card.') exit()

# Infinite Face Detection Loop v\_cap = cv2.VideoCapture(0)

v\_cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, image\_size) v\_cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, image\_size) flag = False

face\_results = [] start = time.time() while (True):

time\_elapsed = time.time() - prev break\_time = time.time() - start ifbreak\_time> 10:

break

ret, frame = v\_cap.read()

iftime\_elapsed> 1. / frame\_rate: # Collect frames every 1/frame\_rate of a second prev = time.time()

frame\_ = Image.fromarray(frame) frames.append(frame\_)

batch\_boxes, prob, landmark = mtcnn.detect(frames, landmarks=True)

frames\_duplicate = frames.copy() boxes.append(batch\_boxes) boxes\_duplicate = boxes.copy()

# show imgs with bbxs face\_results.append(show\_images(frames\_duplicate, boxes\_duplicate, bbx\_color)) frames = []

boxes = []

if cv2.waitKey(1) & 0xFF == ord('q'):

break v\_cap.release()

cv2.destroyAllWindows()

accuracy = (sum(face\_results) / len(face\_results)) \* 100 print('Percentage match ' + '{:.2f}'.format(accuracy))

if accuracy > 10: print('Authorization Successful') returnrender\_template('verify.html')

#verify() #validate()

#return render\_template('Authorization.html')

else:

print('Authorization Unsuccessful') returnrender\_template('Unauthorization.html') quit()

'''@app.route('/verify', methods=["POST"]) def verify():

email = request.form['email']

current\_otp = sendEmailVerificationRequest(

receiver=email) # this function sends otp to the receiver and also returns the same otp for our session storage

session['current\_otp'] = current\_otp returnrender\_template('verify.html') '''

@app.route('/validate', methods=["POST"]) def validate():

# Actual OTP which was sent to the receiver current\_user\_otp = session['current\_otp'] print("Current User OTP", current\_user\_otp)

# OTP Entered by the User user\_otp = request.form['otp'] print("User OTP : ", user\_otp)

ifint(current\_user\_otp) == int(user\_otp): returnrender\_template('voting.html') else:

return "<h3> Oops! Email Verification Failure, OTP does not match. </h3>" @app.route('/add',methods=['POST','GET'])

def add():

ifrequest.method == 'POST':

cname = request.form.get('name')

sql = 'SELECT \* FROM `counts` WHERE `name` = %s' val = (cname,)

mycursor.execute(sql, val)

result = mycursor.fetchall() if result:

for row in result:

num = int(row[2]) print(num)

sql1 = 'UPDATE `counts` SET `count` = %s WHERE `name` = %s' val1 = (num+1, cname)

mycursor.execute(sql1, val1) mydb.commit()

print(num + 1)

returnrender\_template('success.html',msg = 'Vote Added Successfully') else:

return 'No Data' @app.route('/count') def count():

sql = "SELECT \* FROM `counts`" mycursor.execute(sql)

result = mycursor.fetchall() returnrender\_template('count.html',data=result) @app.route('/view')

def view():

sql = "SELECT \* FROM facetb" mycursor.execute(sql)

result = mycursor.fetchall() returnrender\_template('view.html',data=result)

if \_\_name == ' main ':

app.run(debug=True,) import random importsmtplib

defgenerateOTP(otp\_size = 6):

final\_otp = ''

fori in range(otp\_size):

final\_otp = final\_otp + str(random.randint(0,9)) returnfinal\_otp

def sendEmailVerificationRequest(sender="[sdprotrichy2k23@gmail.com](mailto:sdprotrichy2k23@gmail.com)",receiver="t [amilstark17@gmail.com](mailto:amilstark17@gmail.com)", custom\_text="Hello, Your OTP is "):

server = smtplib.SMTP('smtp.gmail.com',587) server.starttls()

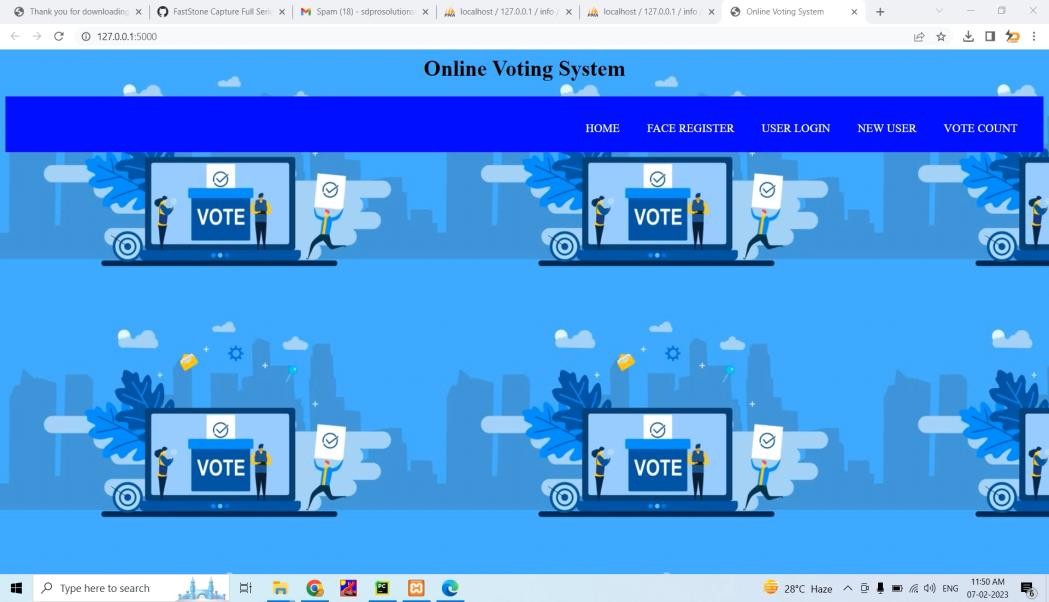
google\_app\_password = "xwycjezbamzaroti" server.login(sender,google\_app\_password) cur\_otp = generateOTP()

msg = custom\_text + cur\_otp server.sendmail(sender,receiver,msg) server.quit()

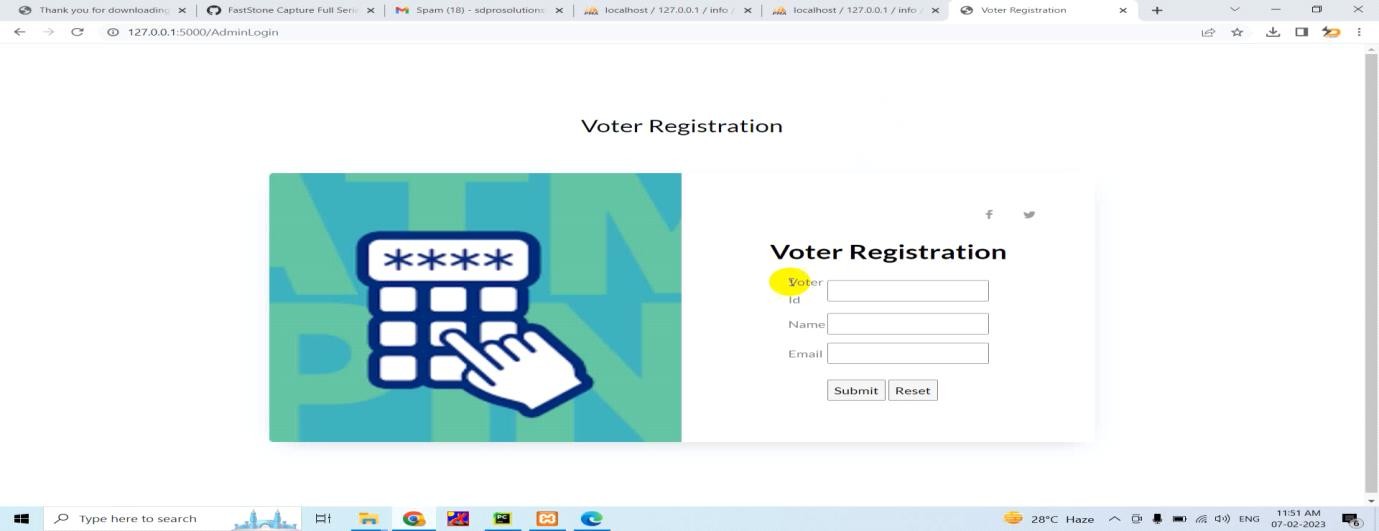
returncur\_otp

## APPENDIX:2

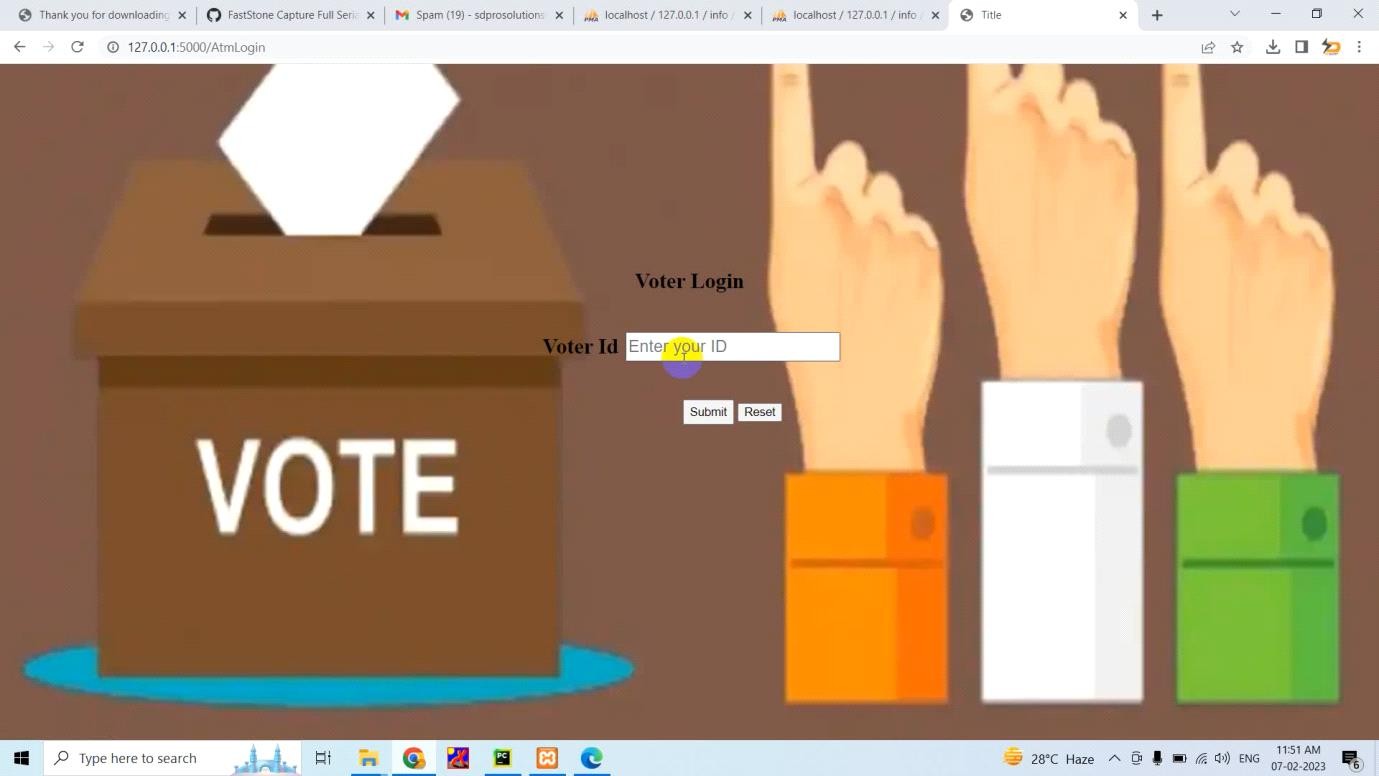
**SCREENSHOTS**



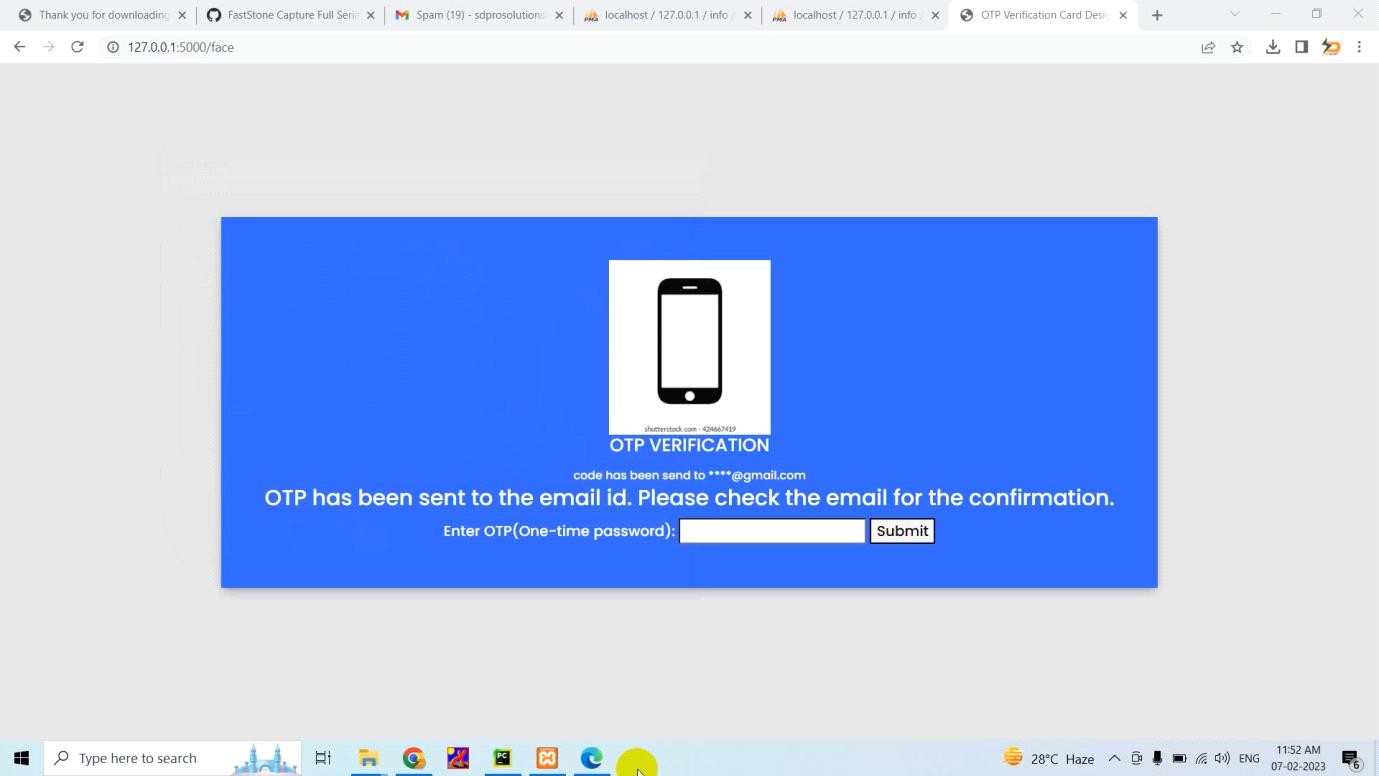
## Figure A.2.1 Home Page



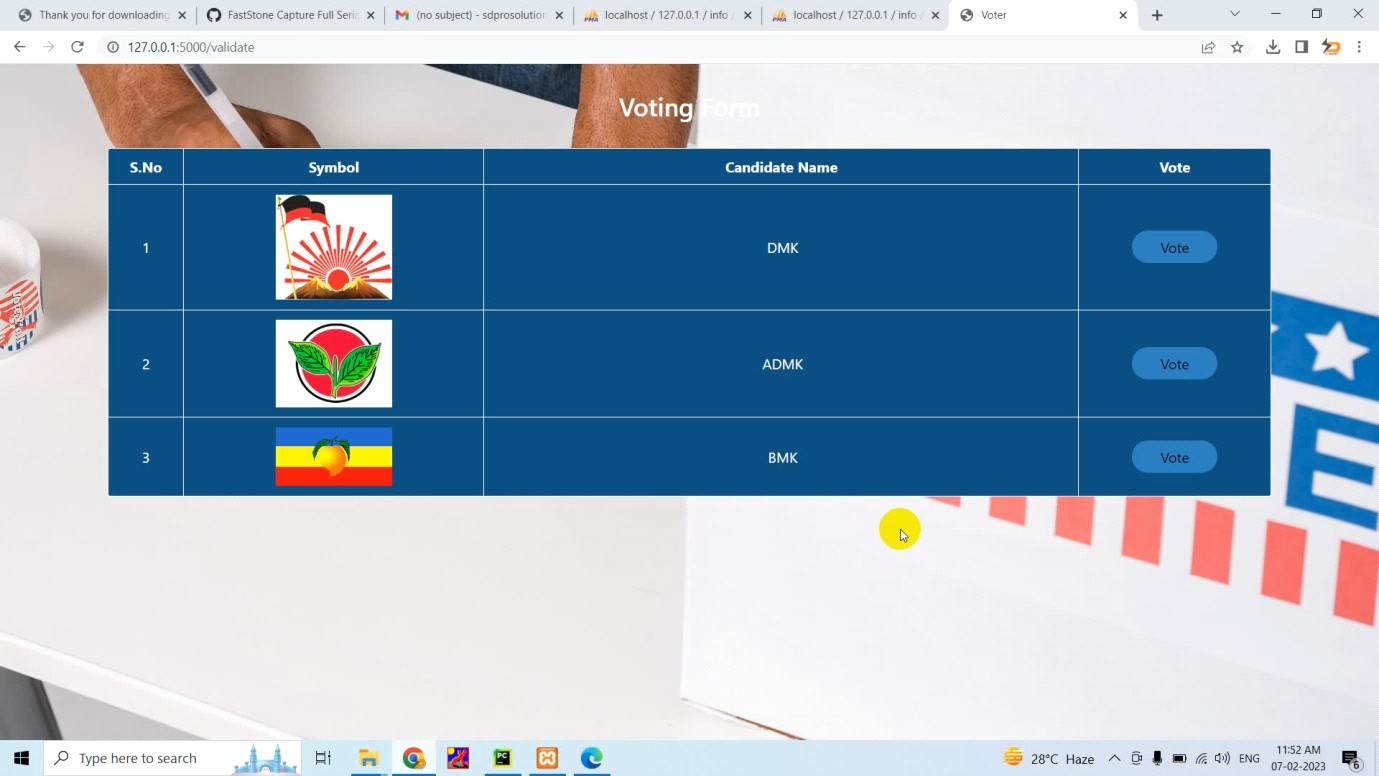
**Figure A.2.2 Voter Registration**



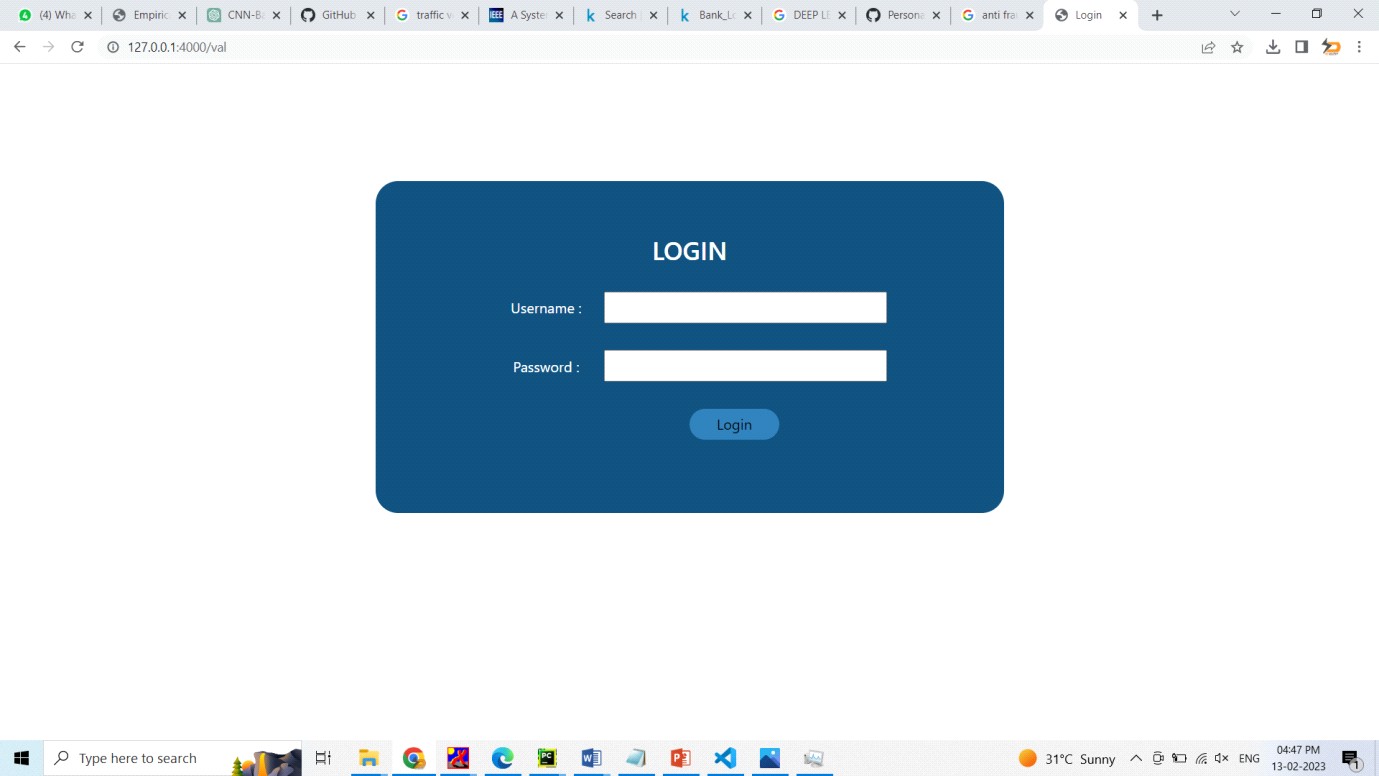
## Figure A.2.3 Voter Login



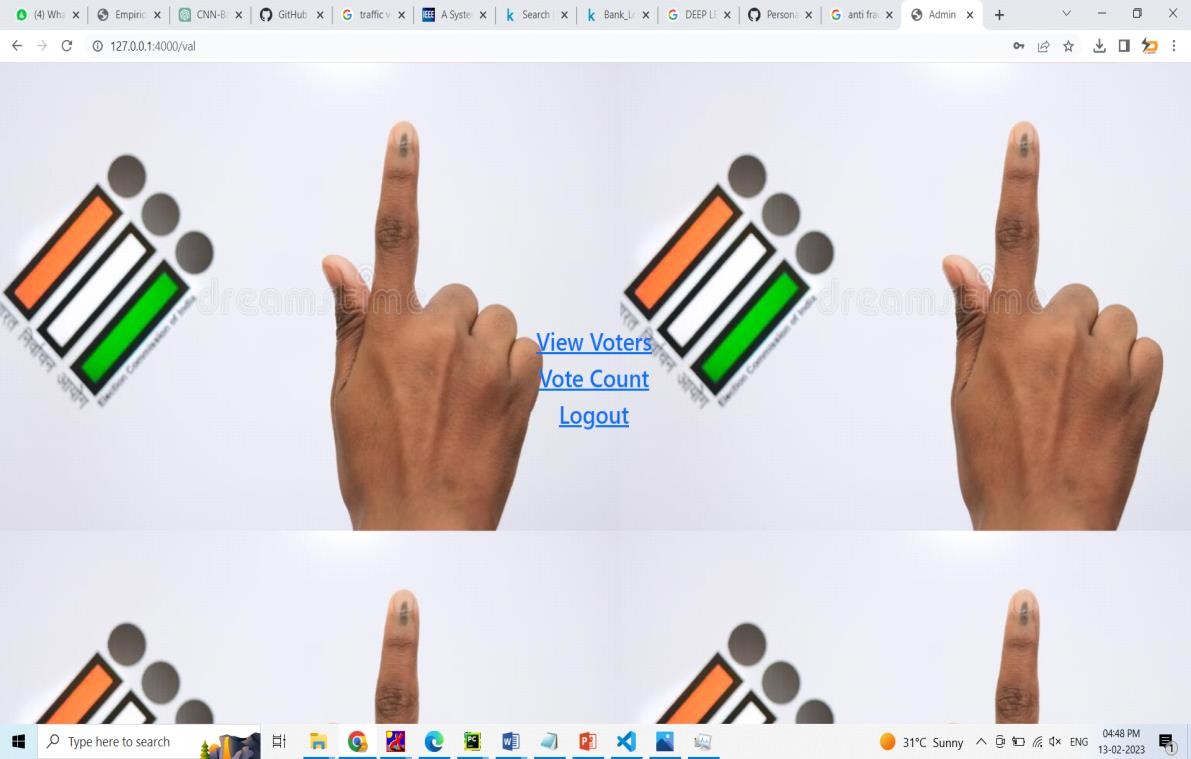
**Figure A.2.4 OTP Verification**



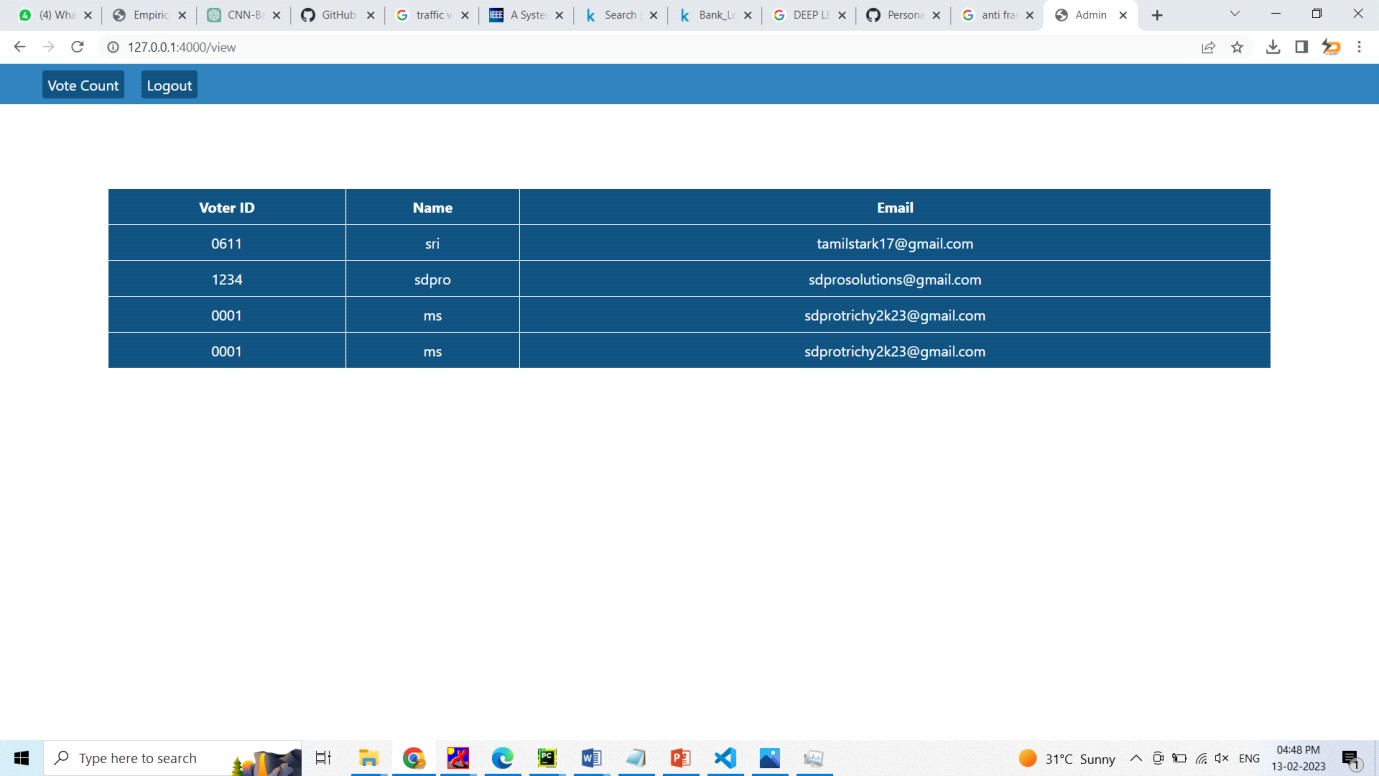
## Figure A.2.5 Add vote



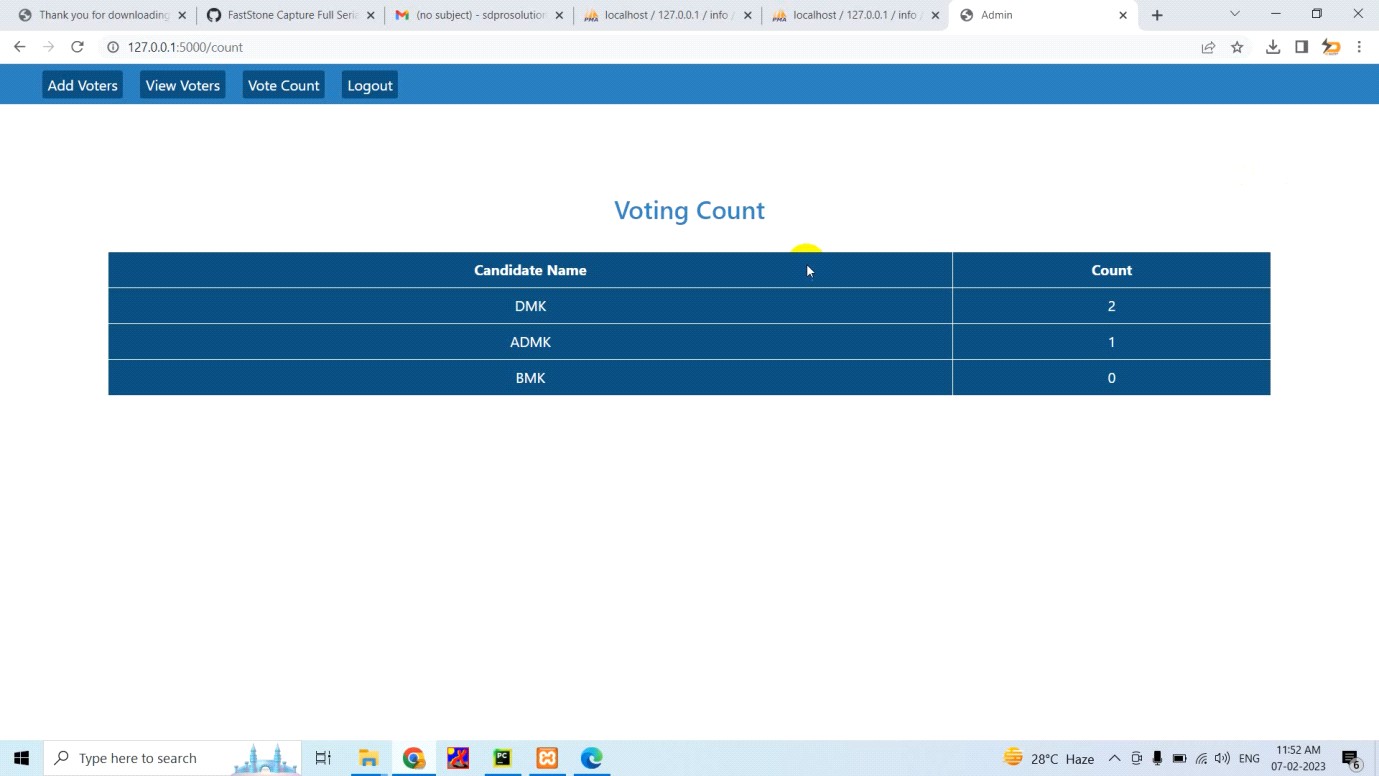
**Figure A.2.6 Admin Login**



## Figure A.2.7 Admin Page



**Figure A.2.8 View Voters**



## Figure A.2.9 Voting Count

**REFERENCE**

1. P. Lombardi, S. Giordano, H. Farouh, and W. Yousef, ‘‘Modelling the smart city performance,’’ Innov., Eur. J. Social Sci. Res., vol. 25, no. 2, pp. 137–149, Jun. 2012.
2. P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, ‘‘Current trends in smart city initiatives: Some stylised facts,’’ Cities, vol. 38, pp. 25–36, Jun. 2014.
3. Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, ‘‘Internet-of-Things-based smart cities: Recent advances and challenges,’’ IEEE Commun. Mag., vol. 55, no. 9, pp. 16–24, Sep. 2017.
4. M. E. M. Cayamcela and W. Lim, ‘‘Artificial intelligence in 5G technology: A survey,’’ in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2018, pp. 860–865.
5. F. Al-Turjman, ‘‘5G-enabled devices and smart-spaces in social-IoT: An overview,’’ Future Gener. Comput. Syst., vol. 92, pp. 732–744, Mar. 2019.
6. R. Molina-Masegosa and J. Gozalvez, ‘‘LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicleto-everything communications,’’ IEEE Veh. Technol. Mag., vol. 12, no. 4, pp. 30–39, Dec. 2017.
7. P. Tarasov and H. Tewari, ‘‘The future of E-voting,’’ IADIS Int. J. Comput. Sci. Inf. Syst., vol. 12, no. 2, pp. 1–19, 2017.
8. N. Kshetri and J. Voas, ‘‘Blockchain-enabled E-voting,’’ IEEE Softw., vol. 35, no. 4, pp. 95–99, Jul. 2018.
9. J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, ‘‘A survey of blockchain technology applied to smart cities: Research issues and challenges,’’ IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.
10. A. A. Monrat, O. Schelén, and K. Andersson, ‘‘A survey of blockchain from the perspectives of applications, challenges, and opportunities,’’ IEEE Access, vol. 7, pp. 117134–117151, 2019.
11. K. Curran, ‘‘E-voting on the blockchain,’’ J. Brit. Blockchain Assoc., vol. 1, no. 2, pp. 1–6, Dec. 2018.
12. M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, ‘‘IoT malicious traffic identification using wrapper-based feature selection mechanisms,’’ Comput. Secur., vol. 94, Jul. 2020, Art. no. 101863.
13. M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, ‘‘A machine learning approach for feature selection traffic classification using security analysis,’’ J. Supercomput., vol. 74, no. 10, pp. 4867–4892, Oct. 2018.
14. M. A. Khan and K. Salah, ‘‘IoT security: Review, blockchain solutions, and open challenges,’’ Future Gener. Comput. Syst., vol. 82, pp. 395–411, May 2018.
15. U. D. Gandhi, P. M. Kumar, R. Varatharajan, G. Manogaran, R. Sundarasekar, and S. Kadu, ‘‘HIoTPOT: Surveillance on IoT devices against recent threats,’’ Wireless Pers. Commun., vol. 103, no. 2, pp. 1179–1194, Nov. 2018.
16. G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, ‘‘A blockchain framework for securing connected and autonomous vehicles,’’ Sensors, vol. 19, no. 14, pp. 1–15, 2019.



