

A
MAJOR PROJECT REPORT ON
**SMART VOTING SYSTEM THROUGH FACE
RECOGNIZATION WITH OTP**

Submitted in partial fulfillment of the requirement for the award of degree of
BACHELOR OF TECHNOLOGY

IN
ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY:

M PRASANNA KUMAR	218R1A04A0
M DINESH	218R1A04A1
M NAGA RAKSHITHA	218R1A04A2
MD AWAIS HUSSAIN	218R1A04A3

Under the Esteemed Guidance of

Dr. T.SATYANARAYANA
Associate professor



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)
Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya (V), Medchal Road, Hyderabad - 501401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING



CERTIFICATE

This is to certify that Major project work entitle “**SMART VOTING SYSTEM THROUGH FACE RECOGNIZATION WITH OTP**” submitted by **M PRASANNA KUMAR** bearing Roll No: **218R1A04A0**, **M DINESH** bearing Roll No: **218R1A04A1**, **M NAGA RAKSHITHA** bearing Roll No: **218R1A04A2**, **MD AWAIS HUSSAIN** bearing Roll No: **218R1A04A3** in Batch IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out by then during the other University for the award of any degree.

INTERNAL GUIDE

Dr. T.SATYANARAYANA

HEAD OF DEPARTMENT

Dr. SUMAN MISHRA

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our Principal **Dr. A. S. Reddy** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA**, Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr.SATYANARAYANA** , Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, through out the course of this work. We sincerely thank our project internal guide **Dr.SATYANARAYANA**, Associative Professor of ECE for guidance and encouragement in carrying out this project work

DECLARATION

We hereby declare that an major-project entitled “**SMART VOTING SYSTEM THROUGH FACE RECOGNIZATION WITH OTP**” is the work done by us in campus at **CMR ENGINEERING COLLEGE**, Kandlakoya during the academic year 2024-2025 and is submitted as major project in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

M PRASANNA KUMAR

218R1A04A0

M DINESH

218R1A04A1

M NAGA RAKSHITHA

218R1A04A2

MD AWAIS HUSSAIN

218R1A04A3

ABSTRACT

Some forms of voting have been here ever since. Mostly used form all over the world are paper ballots. Electronic voting schemes are being popular only in the last decade and they are still unsolved. E-voting schemes bring problems mainly regarding security, credibility, transparency, reliability, and functionality. Estonia is the pioneer in this field and may be considered the state of the art. But there are only a few solutions using blockchain. Blockchain can deliver an answer to all of the mentioned problems and furthermore bring some advantages such as immutability and decentralization. The main problems of technologies utilizing blockchain for e- voting are their focus on only one field or lack of testing and comparison. In this paper, we present a blockchain-based e-voting platform, which can be used for any kind of voting. It is fully utilized by blockchain and all processes can be handled within it. After the start of the voting, the platform behaves as fully independent and decentralized without possibilities to affect the voting process. The data are fully transparent, but the identity of voters is secured by homomorphic encryption. We have tested and compared our solution in three different blockchains. The results show, that both public and private blockchains can be used with only a little difference in the speed. The key novelty of our solution is a fully decentralized management of e-voting platform through blockchain, transparency of the whole process and at the same time security and privacy of the voters thanks to homomorphic encryption.

CONTENTS

CHAPTERS	PAGE NO
CHAPTER-1	
INTRODUCTION	
1.1 OVERVIEW OF THE PROJECT	1
1.2 OBJECTIVE OF THE PROJECT	2
1.3 ORGANIZATION OF THE PROJECT	3
CHAPTER-2 LITERATURE SURVEY	
2.1 BLOCKCHAIN-ENABLD E-VOTING	4
2.2 VOTING PROCESS WITH BLOCKCHAIN TECHNOLOGY: AUDITABLE BLOCKCHAIN VOTING SYSTEM	4
2.3 A SMART CONTRACT FOR BOARD ROOM VOTING WITH MAXIMUM VOTER PRIVACY	5
2.4 DEFINITIONS AND PROPERTIES OF ZERO KNOWLEDGE PROOF SYSTEM	6-7
2.5 SOFTWARE ENVIRONMENT	7-8
2.6 WHY CHOOSE PYTHON	8-12
CHAPTER-3 SYSTEM ANALYSIS	
3.1 EXISTING SYSTEM	13-15
3.2 PROPOSED SYSTEM	15-19
CHAPTER-4 FEASIBILITY STUDY	
4.1 ECONOMICAL FEASIBILITY	20
4.2 TECHNICAL FEASIBILITY	20
4.3 SOCIAL FEASIBILITY	20-21
4.4 ENVIRONMENTAL FEASIBILITY	21
4.5 OPERATIONAL FEASIBILITY	21
4.6 MARKET FEASIBILITY	21
CHAPTER-5 SYSTEM REQUIREMENTS	
5.1 HARDWARE REQUIREMENTS	22
5.2 SOFTWARE REQUIREMENTS	22

CHAPTER-6	23-29
SYSTEM DESIGNS	
SYSTEM IMPLEMENTATIONS	30-37
SYSTEM TESTING	38-42
OUTPUT SCREENS	43-52
FUTURE ENHANCEMENT	53
CONCLUSION	54
REFERENCE	55
APPENDIX	56

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	System Architecture online voting system using facial recognition	1
1.2	Block diagram of image processing	2
1.3	Face Recognition presentation	3
2.1	Voting Process with Blockchain	4
2.2	Smart id management system	5
2.3	Software Environment	7
2.4	Python Development benefits	10
3.1	Admin Login	18
3.2	Username and Password Screen	19
3.3	Add party Details	19
3.4	Add Candidate details	20
3.5	Adding other party details	20
3.6	Displaying party details	21
3.7	Registration page	21
3.8	Sign-Up screen	22
3.9	User Login Screen	22
3.10	Cast your Vote	23
3.11	E-Voting take snapshot	23
3.12	Validate User	24
3.13	Displaying the list of candidates	24
3.14	Vote Accepted	25
3.15	Re-casting the vote	25
3.16	Already Casted vote screen	26
3.17	Admin Login screen	26
3.18	View Votes	27
3.19	Vote count display screen	27

CHAPTER-1

INTRODUCTION

1.1 Overview of the project

The topic of e-voting systems is still at an early stage of development. We have chosen this domain not only for its recency but also because there are not many solutions that address problems of e-voting. Nowadays, popularity grows also in the development of e-Government. However, such a system is not feasible if basic services for citizens such as elections do not become electronic. "E-voting is one of the key public sectors that can be transformed by blockchain technology". Hand by hand with e-voting come also new challenges, which need to be addressed. One of them is e.g. securing the elections, which needs to be at least as safe as the classic voting systems with ballots. That is why we have decided to create safe elections in which voters do not have to worry about someone abusing the electoral system.

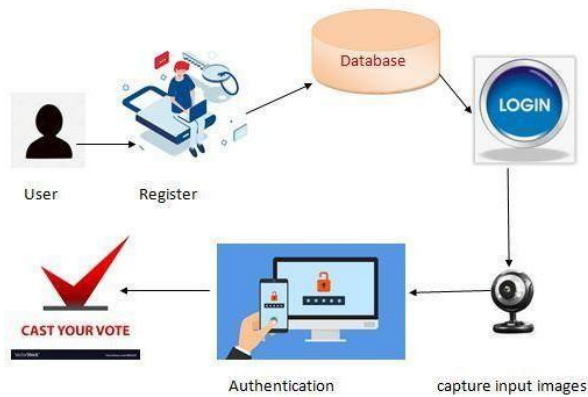


Fig 1.1: System Architecture online voting system using facial recognition

In recent years blockchain is often mentioned as an example of secure technology used in an online environment. Our e-voting system uses blockchain to manage all election processes. Its main advantage is that there is no need for confidence in the centralized authority that created the elections. This authority cannot affect the election results in our system. Another challenge in e-voting is the lack of transparency in the functioning of the system, leading to a lack of confidence in voters [2]. This problem is solved by blockchain in a way of total transparency that allows everyone to see the stored data and processes such as how are these data handled. In the field of security, this technology is more suitable in every way than the classic e-voting platform without blockchain.

1.2 Objective of the project

The Objective of the project that the election plays an important role in such a huge democratic country like India where the leader is elected by residents. Elections preserve a truthful state functioning, as they provide people the choice to select their personal government. So the election ought to be an unfastened and truthful process. Every citizen of a democratic country has a right of voting with his/her own choice. One of the fundamental issues in the conventional democratic framework is that it expends bunches of labor and resources. Also some humans can be worried about illegal publications of movement at some point of this manner of election or its preparation. There are some disadvantages of the conventional election voting process which is being used in our country such as machine stops working, chances of brutality, time consuming, resource consuming, spot arranged etc. Many people couldn't vote because the voter has to reach the poll booths to vote or some people like those who are living far away from their original birth place where they are allowed to vote. So to get rid of their drawbacks, a new System is introduced i.e. Online Voting System, which provides

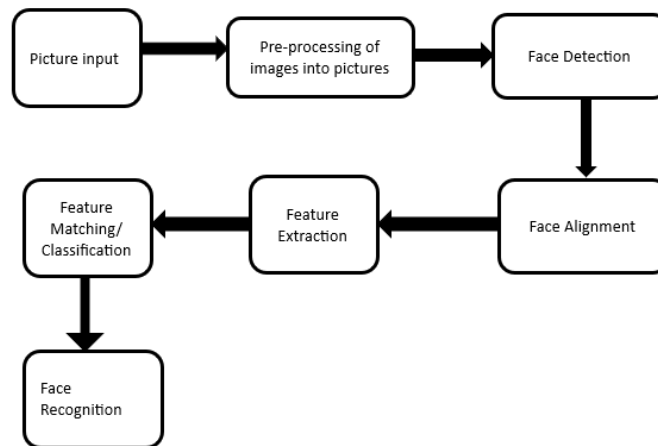


Fig 1.2: Block Diagram of image processing

accuracy, security, flexibility, mobility etc. An online voting System in a web based application to use in the election process. Initially ballot paper technique was used in the election process. Then the Electronic Voting Machine comes, these are easy to store the data and easily manageable.

These are more secure than the ballot paper and less time consuming. Now, we proposed a system with biometric authentication to make the voting process more

secure and reduce the time taken in the voting process. By the use of this, the electorate can solidify their vote for his or her preferred candidate through the use of their system.

Organization of the project

We use Face detection and Recognition Technology for authentication of citizens that he/she is the proper consumer or not. We provide many modules in which admin can login withinside the tool and show the numerous operations. Also users can login in the system and use their right to vote. When the Voter uses the system, the system will capture his/her image using a web camera & try to match with the image stored in the database. If both images are the same then the voter can cast his/her vote. Most higher learning establishments in Kenya conduct elections routinely to elect an understudy leadership to choose them. They proposed the process of an online system, which includes systems like enlistment of voters, vote casting, vote checking, and pronouncing results which would establish a decent answer for a substitute of framework that is in the institutes in Kenya.

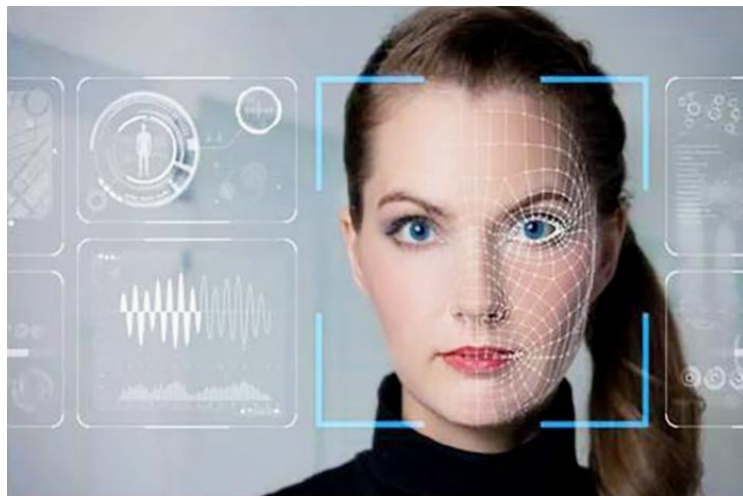


Fig 1.3: Face Recognition Presentation

Aims and scope objectives

The aim is to develop an application that seeks to use various stages of security authentication to enhance the election process for political party elections using the real case study, i.e. The University of Ibadan, in the end imparting an internet platform which permits all eligible electorate to workout their franchise from any region for the duration of the election period. The targets are:

1.To create a secured online voting platform where authenticity of votes and voters are ensured with the use of mechanisms such as facial recognition and one – time password

2.To enhance Voter's identity due to the fact that biometric functions can not be shared.

CHAPTER-2

LITERATURE SURVEY

2.1 Blockchain-enabled e-voting

Authors: N.Kshetri and J.Voas

Abstract: Blockchain-enabled e-voting (BEV) could reduce voter fraud and increase voter access. Eligible voters cast a ballot anonymously using a computer or smartphone. BEV uses an encrypted key and tamper-proof personal IDs. This article highlights some BEV implementations and the approach's potential benefits and challenges.

2.1 Voting process with blockchain technology: Auditable blockchain voting system

Authors: M. Pawlak, J. Guziur, and A. Poniszewska-Maranda,

Abstract: There are various methods and approaches to electronic voting all around the world. Each is connected with different benefits and issues. One of the most important and prevalent problems is lack of auditing capabilities and system verification methods. Blockchain technology, which recently gained a lot of attention, can provide a solution to this issue. This paper presents Auditable Blockchain Voting System (ABVS), which describes e-voting processes and components of a supervised internet voting system that is audit and verification capable. ABVS achieves this through utilization of blockchain technology and voter-verified paper audit trail.

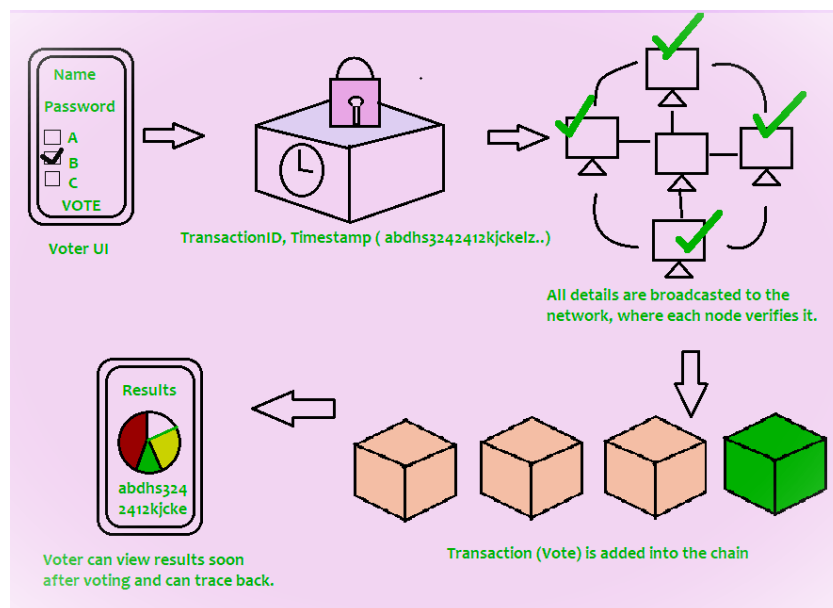


Fig 2.1: Voting process with Blockchain

2.3 A smart contract for boardroom voting with maximum voter privacy

AUTHORS: P. McCorry, S. F. Shahandashti, and F. Hao,

Abstract: We present the first implementation of a decentralized and self-tallying internet voting protocol with maximum voter privacy using the Blockchain. The Open Vote Network is suitable for boardroom elections and is written as a smart contract for Ethereum. Unlike previously proposed Blockchain e-voting protocols, this is the first implementation that does not rely on any trusted authority to compute the tally or to protect the voter's privacy. Instead, the Open Vote Network is a self tallying protocol, and each voter is in control of the privacy of their own vote such that it can only be breached by a full collusion involving all other voters. The execution of the protocol is enforced using the consensus mechanism that also secures the Ethereum blockchain. We tested the implementation on Ethereum's official test network to demonstrate its feasibility. Also, we provide a financial and computational breakdown of its execution cost. We argue that all known zero-knowledge proofs are in fact black box-simulation zero-knowledge (i.e., we proved zero-knowledge using black box-simulation of the verifier).

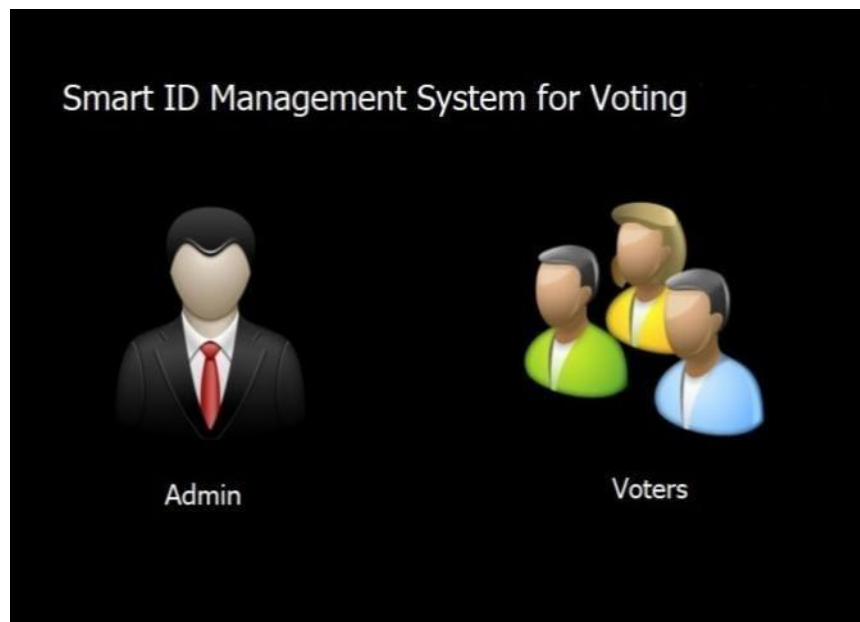


Fig 2.2: Smart ID management system

2.4 Definitions and properties of zero knowledge proof systems

AUTHORS:O. Goldreich and Y. Oren,

Abstract: In this paper we investigate some properties of zero-knowledge proofs, a notion introduced by Goldwasser, Micali, and Rackoff. We introduce and classify two definitions of zero-knowledge: auxiliary-input zero-knowledge and blackbox-simulation zero-knowledge. We explain why auxiliary-input zero-knowledge is a definition more suitable for cryptographic applications than the original [GMR1] definition. In particular, we show that any protocol solely composed of subprotocols which are auxiliary-input zero-knowledge is itself auxiliary-input zero-knowledge. We show that blackbox-simulation zero-knowledge implies auxiliary-input zero-knowledge (which in turn implies the [GMR1] definition). We argue that all known zero-knowledge proofs are in fact black box-simulation zero-knowledge (i.e., we proved zero-knowledge using black box-simulation of the verifier). As a result, all known zero-knowledge proof systems are shown to be auxiliary-input zero-knowledge and can be used for cryptographic applications such as those in [GMW2]. We demonstrate the triviality of certain classes of zero-knowledge proof systems, in the sense that only languages in BPP have zero-knowledge proofs of these classes. In particular, we show that any language having a Las Vegas zero-knowledge proof system necessarily belongs to RP. We show that randomness of both the verifier and the prover, and nontriviality of the interaction are essential properties of (nontrivial) auxiliary-input zero-knowledge proofs.

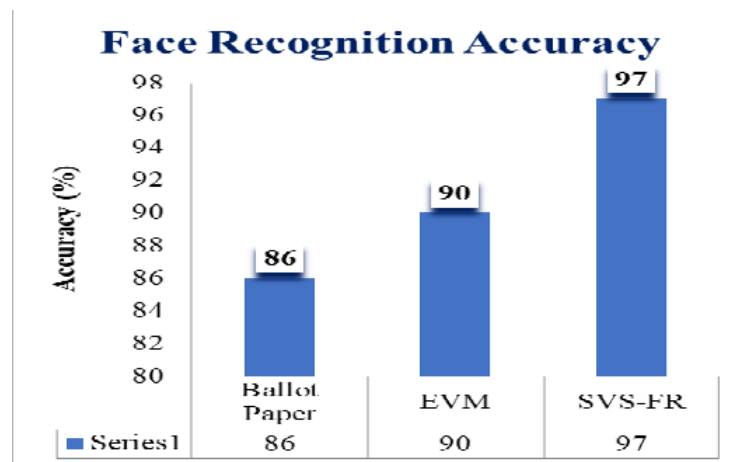


Fig 2.3.1: Face recognition graph

The proposed system in the paper has 3 security phases. Information of individuals above age 18 will be taken from the Aadhar database. In the first phase, the voters will be given an Id and password through the registered e-mail Id before the voting process.

[3] The second phase is validating the voter using fingerprints data and once confirmed voter will be allowed to cast the vote. After casting, as a part of the third phase, the voter id will be deleted leaving no second chance to vote again. Aadhar details that were used by the voter will be locked to track the voter for further access. The count will be updated parallel.

2.1 Software environment

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Source in 1991, and version 1.0 was released in 1994.

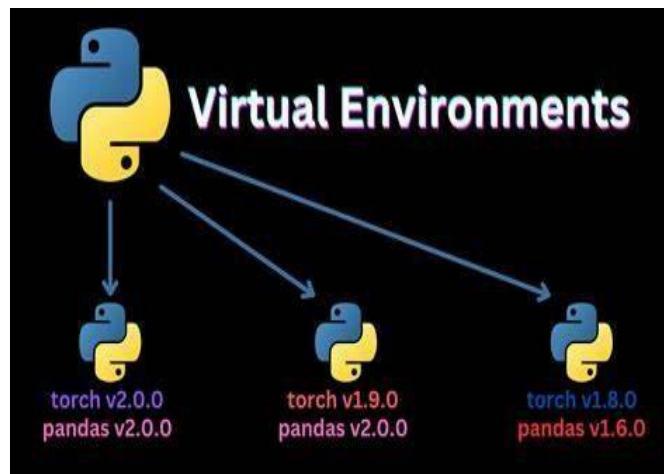


Fig 2.1.1 Python versions

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do. Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

2.2 Why choose python

If you're going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day. According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index.

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like to dig deeper into Python salary statistics and job opportunities, you can do so [here](#).

Python is interpreted

Any languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly. This makes for a quicker development cycle because you just

type in your code and run it, without the intermediate compilation step one potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect. Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It's Not That Simple

For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming. Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming. Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Conclusion

This section gave an overview of the **Python** programming language, including:

- A brief history of the development of Python
- Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

Python is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the

results. This also means Python is slower than a compiled language like C, because it is not running machine code directly. Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages. Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp. The server goes through all the images in the database and tries to find a match in the registered ones. If a matching face is found, the voter is registered and recognized by the election commission and hence is allowed to vote. If a match is not found, the page will say he or she is not recognized and would not be allowed to vote. The main security level is where the system recognizes the face of the voter from the current database of face images given by the election commission. If the image captured matches the respective image of the voter in the database, then a voter can cast their vote in the election. Haar Cascade Algorithm is used to extract the facial features and to recognize the facial part of the image. Visual Studio and software like Python, HTML, and Django were used to create the online platform and to implement the algorithm.

Python Development Benefits



Fig 2.4: Python Development Benefits

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses

Python use [change / change source]

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

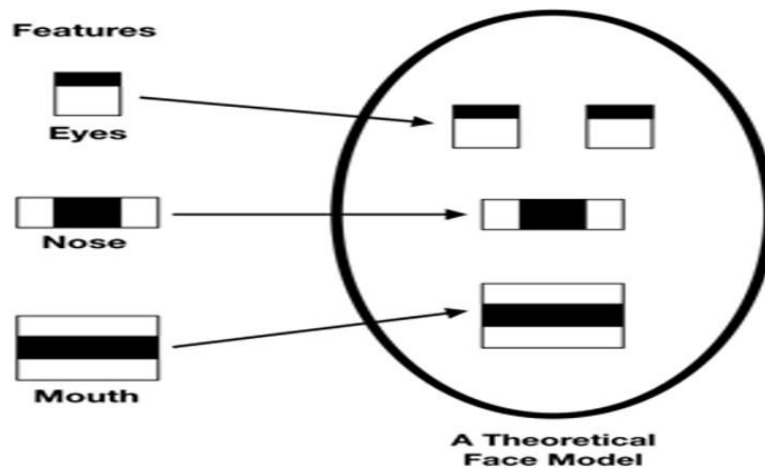


Fig 2.5 Face model task

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

CHAPTER-3

SYSTEM ANALYSIS

3.1 Existing system:

In recent years blockchain is often mentioned as an example of secure technology used in an online environment. Our e-voting system uses blockchain to manage all election processes. Its main advantage is that there is no need for confidence in the centralized authority that created the elections. This authority cannot affect the election results in our system. Another challenge in e-voting is the lack of transparency in the functioning of the system, leading to a lack of confidence in voters.

The existing system is not too effective. At present there are two types of voting methods, they are:

- Ballot Voting
- EVM Voting
- Ballot Voting

A ballot is a device used to cast votes in an election, and may be a piece of paper used in secret voting. In this the voter is given a paper which consists of all the party symbols along with representative names in it. Here, people come to the polling booth, take the ballot paper and vote by putting a stamp on the desired party symbol. Finally, the ballot paper is folded and dropped into the ballot box. At last, the votes are counted by the Election commission officers.

EVM (Electronic Voting Machine) Voting

An EVM is a device which is used for voting. This machine consists of party symbols along with the representative's name and a button at the end for each and every party name. The voters come near the EVM machine after completion of their verification at the early level before voting. After verification the voter goes near the EVM and casts their vote by pressing the button. The above procedures are not so accurate as there may be possibility for the false/fake voting. The ballot papers may be lost at the time of counting which may affect results of the particular area or people may miscount the number of votes which leads authority into wrong hands. EVM machines sometimes get corrupted and polling gets stopped temporarily and a lot of time is wasted or EVM may be tampered and the casted votes may be polled to a

particular party only, even the vote is casted to different candidates or parties. This may lead authority into the wrong hands. They also lack security as one's vote can be casted by another voter or even a miscellaneous person. This factor is known as fake voting. Without proper authentication there is a possibility of fake voting. So, the existing system is not efficient for voting. Even though there is very little false/fake voting, this minor setback can turn the results in the opposite direction.

Related work

Several authors have put in efforts in the field of face recognition, significant contributions are briefed in the literature review. Vigorous technique for naturally coordinating highlights in pictures compared to the equivalent physical point on an item observed from two discretionary perspectives. Unlike conventional stereo matching approaches coordinating methodologies, the presumption like no earlier information about the relative camera positions and directions. Actually in this application this is the data wish to decide from the picture feature .matches. Highlights are distinguished in two or more pictures and portrayed utilizing affine texture invariants. The fundamental test is the way to improve the recognition performance when influenced by the fluctuation of non-linear effects that incorporate illumination variances, poses, facial expressions, occlusions and so on. A robust 4-layer Convolutional Neural Network (CNN), engineering is proposed for the face acknowledgment issue, with an answer that is equipped for dealing with facial pictures that contain occlusions, poses, facial expressions. There are many face recognition algorithms, just a bunch of them meet the continuous limitations of a software based arrangement without utilizing any committed hardware engine. This paper presents a real-time and robust solution for mobile platforms, which in general have limited computation and memory resources as compared to PC platforms. This solution includes joining two previous real-time implementations for mobile platforms to address the shortcoming of each implementation. The main execution gives an on the web or on-the-fly light source adjustment for the second usage which is seen as robust to various face postures or orientations. Pattern classification approach by considering every pixel in an image as a coordinate in a high-dimensional space.

Along with the upside of the perception that the images of a specific face, under fluctuating illumination but fixed pose, lie in a 3D linear subspace of the high

dimensional image space—if the face is a Lambertian surface without shadowing. In any case, since faces are not genuinely Lambertian surfaces and in fact produce self-shadowing; images will deviate from this linear subspace. As opposed to explicitly projecting this deviation, linearly project the image into a subspace in a way which limits those regions of the face with huge deviation. Design and execution of the component extraction strategy for Speeded-Up Robust Features (SURF) and Support Vector Machine (SVM) grouping technique into the traffic signs recognition application. The yield of this application is the importance of the traffic sign with two languages, Indonesia and English. In the SURF strategy, the littlest huge number of key focuses will influence the accuracy level to perceive a picture. Face detection is the premise of all the face processing systems, while in video the face detection issue has more special importance. By examining the face detection dependent on Adaboost algorithm, this paper presents a quick and good robust face detection method. Firstly, the motion region which contains faces is obtained based on motion detection, excluding the background interference. Secondly, the Adaboost algorithm is used to detect the face in the motion region and locate the face. The experiments show that this method can rapidly and accurately detect human faces. The face recognition and tracking and the advancement of the customer side of the system uses Android cell phones. For the face recognition stage, Viola-Jones algorithm is used that isn't influenced by illuminations. The face tracking stage depends on the Optical Flow algorithm. Optical Flow is implemented in the framework with two component extraction strategies, Fast Corner Features and Regular Features. Real-time robust technique is created to distinguish irises on faces with coronal axis rotation within the normal range. The technique permits head movement without any limitations to the background. The technique depends on anthropometric templates applied to recognize the face and eyes. The template uses key features of the face, for example, elliptical shape, and location of the eyebrows, nose, and lips.

3.1 PROPOSED SYSTEM:

The proposed blockchain voting system considers all requirements for voting and is designed generally for any elections e.g. president, student parliament, etc. The system allows more round elections and preferably uses a public blockchain. The public blockchain can be replaced by other types of blockchain but the stored data (votes) have

to be easily verified by any user. The user represents any observer who is interested in the blockchain voting. In our proposed system we identify three main roles: vote publisher; key authority; and voter. These three roles can represent an organization, a company, or a user. The roles vote publisher and key authority can be grouped to one role due to that they can be the same organization or person. The voter attends the elections depending on vote configuration. The configuration of the votes is performed by the vote publisher and is included in the smart contract. The vote publisher has to know all cipher keys before publishing the smart contract. The close collaboration between the vote publisher and the key authority is required. The key authority creates and distributes all cipher keys to a voter and vote publisher. The distributing channel has to be secured and should not be vulnerable to any 3rd party.

3.3 ADVANTAGES OF PROPOSED SYSTEM:

We can vote through automatic process.it is easy to vote.

System analysis for a smart voting system through face recognition involves examining the functional and non-functional requirements of the system. Here's a breakdown of the analysis:

Functional Requirements

User Registration: Voters must register themselves by providing personal details and facial data.

Face Recognition: The system must accurately recognize and verify the voter's face using facial recognition algorithms ².

Voting Process: The system must allow voters to cast their ballots securely and efficiently.

Result Processing: The system must accurately count and display the voting results.

Non-Functional Requirements

Security: The system must ensure the security and integrity of the voting process, protecting against fraudulent activities and data breaches.

Accuracy: The system must accurately recognize and verify voter faces, ensuring the validity of the voting results.

Efficiency: The system must provide a fast and efficient voting experience for voters.

Scalability: The system must be able to handle a large number of voters and voting

processes.

System Components

Face Recognition Module: Responsible for recognizing and verifying voter faces.

Voting Module: Handles the voting process, including ballot casting and result processing

Database Management System: Stores and manages voter data, voting results, and system configuration.

User Interface: Provides a user-friendly interface for voters to interact with the system.

System Analysis Tools

Use Case Diagrams: Visualize the system's functional requirements and user interactions.

Class Diagrams: Model the system's components and their relationships. **Sequence**

Diagrams: Illustrate the system's interactions and workflows. **State Machine**

Diagrams: Describe the system's states and transitions.

By conducting a thorough system analysis, we can ensure that the smart voting system through face recognition meets the functional and non-functional requirements, providing a secure, efficient, and accurate voting experience for voters.

Input design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES:

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

Output design:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

Select methods for presenting information.

1. Create document, report, or other formats that contain information produced by the system.

2. The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER-4

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

4.1 Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.2 Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.3 Social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of

acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.4 Environmental feasibility

1. Energy Consumption: The system will require energy to operate, which will depend on the hardware and software requirements.
2. E-Waste: The system will require measures to reduce electronic waste and ensure responsible disposal.

4.5 Operational feasibility

1. System Integration: The system will require integration with existing voter registration databases and voting systems.
2. User Training: The users will require training on the system's operation and maintenance.
3. Technical Support: The system will require technical support and maintenance.

4.6 Market feasibility

1. Target Market: The target market will include government agencies, electoral commissions, and private organizations.
2. Market Size: The market size will depend on the adoption rate of the technology.
3. Competitive Analysis: The competitive analysis will include an evaluation of existing voting systems and face recognition technologies.

The Smart Voting System using Face Recognition is technically, financially, and operationally feasible. However, the success of the project will depend on the public acceptance, privacy concerns, and accessibility measures. The project requires a comprehensive plan to address these concerns and ensure the successful implementation of the system.

CHAPTER-5

SYSTEM REQUIREMENTS

5.1 Hardware requirements:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 1 GB

5.2 Software requirements:

- Operating system : Windows 10
- Coding Language : python
- Tool : PyCharm
- Database : MYSQL
- Server : Flask

CHAPTER- 6

SYSTEM DESIGN

6.1 System architecture:

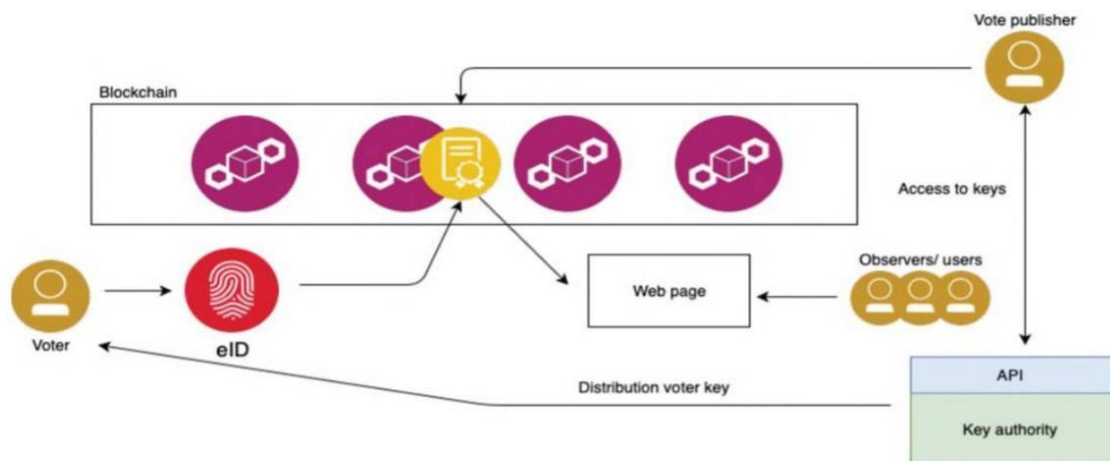


Fig. 1. High-Level architecture

6.2 Data flow diagram:

1. 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.
2. 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.
3. 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.

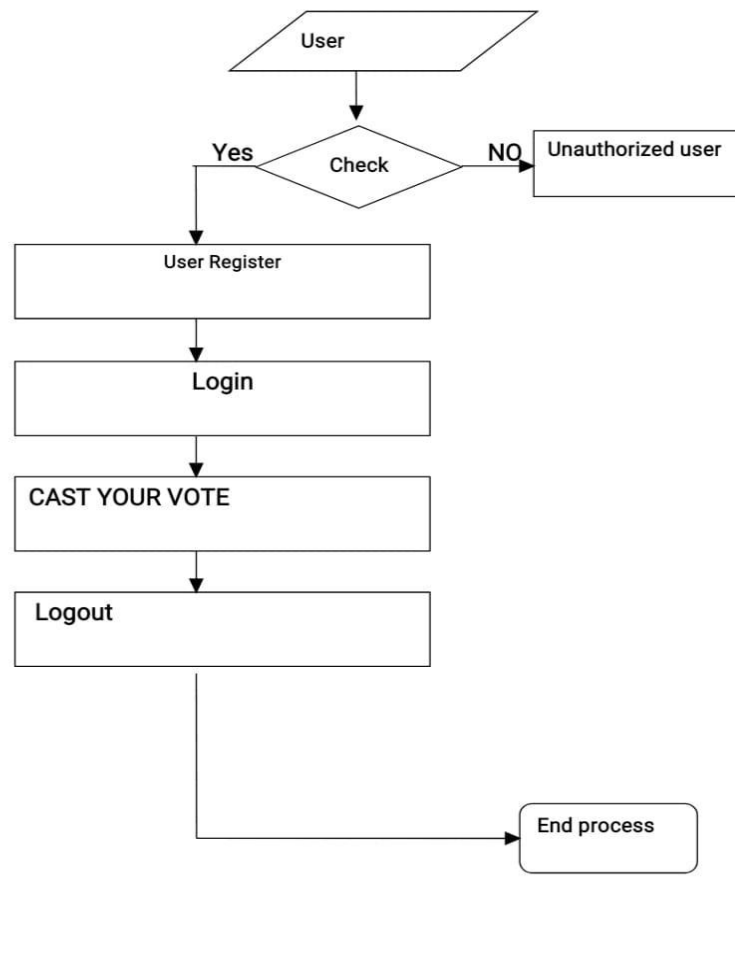


Fig 6.2.1 Admin flow graph

6.3 Uml diagrams:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object- oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the

future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Integrate best practices.

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.

Class diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

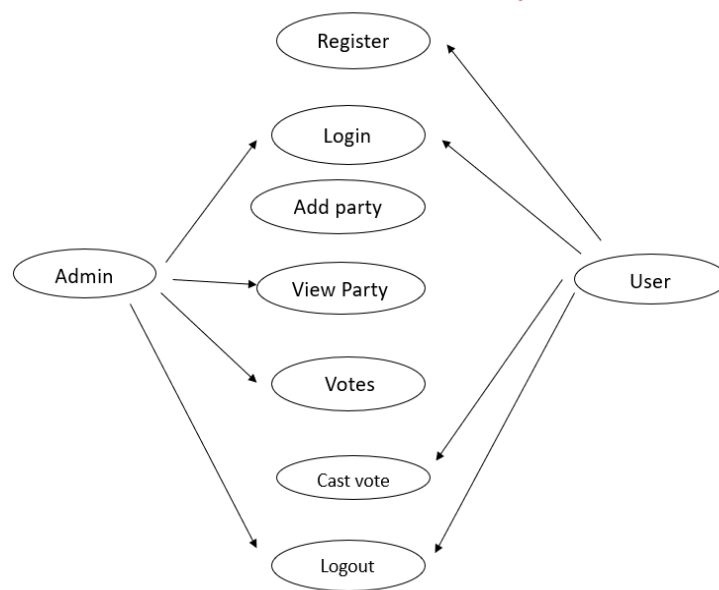


Fig 6.3.1 Sequence diagram

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.

Activity diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

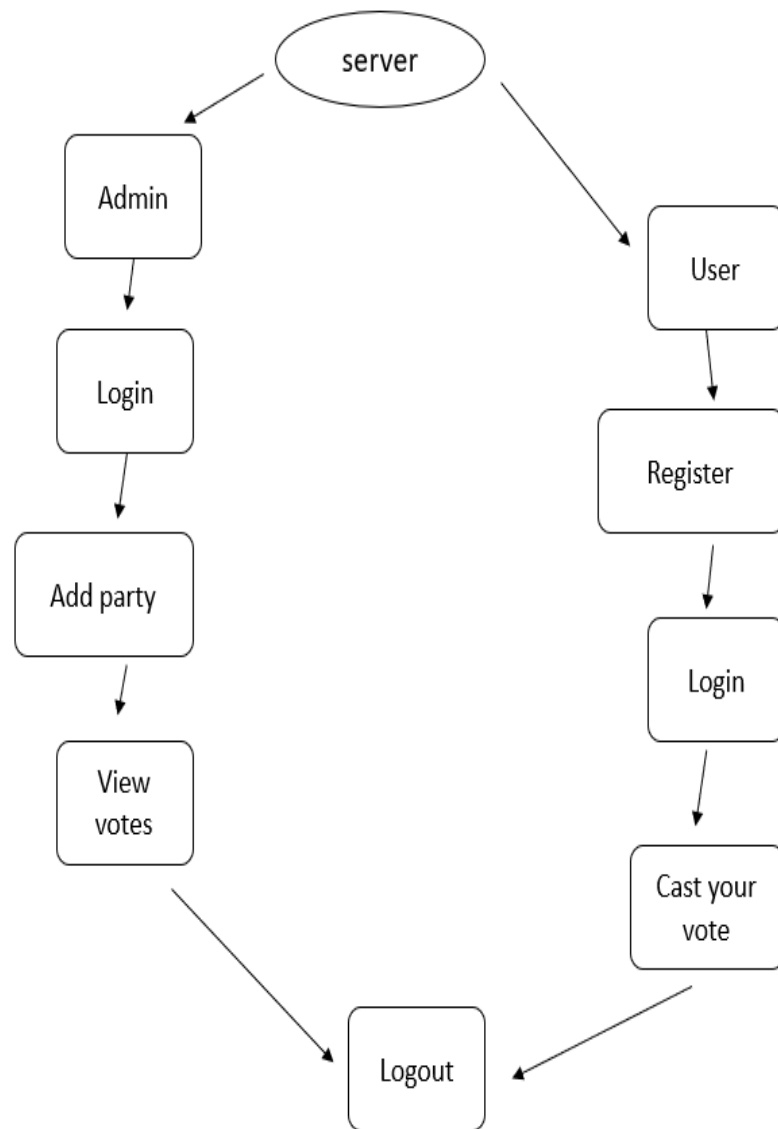


Fig 6.3.2 User flow graph

SYSTEM IMPLEMENTATION

Modules:

- ADMIN
- USER

Modules description:

In this project we are using public python Blockchain API's to store and manage voting data as Blockchain provides secure and tamper proof of data storage and to implement this project we have designed following modules.

Admin module: this user responsible to add new party and candidate details and can view party details and vote count. Admin login to system by using username as 'admin' and password as 'admin'.

User Module: this user has to signup with the application by using username as his ID and then upload his face photo which capture from webcam. After registering user can go for login which validate user id and after successful login user can go for cast vote module which execute following functionality.

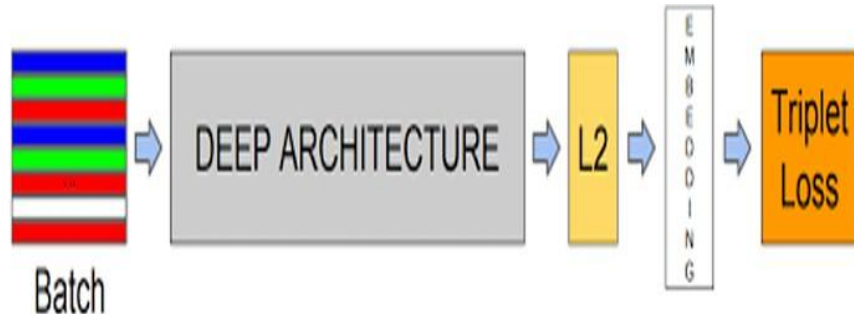


Fig. 5. Image depicting the process in face net algorithm

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.

Types of tests

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.

Features	Proposed Voting System (SVS-FR)	Electronic Voting Machine	Ballot Paper
Security	High	Medium	Low
Reliability	High	High	Medium
Accuracy	High	Medium	Low
Accessibility	High	Medium	Medium
Mobility	High	Low	High

Features of the system

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 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.

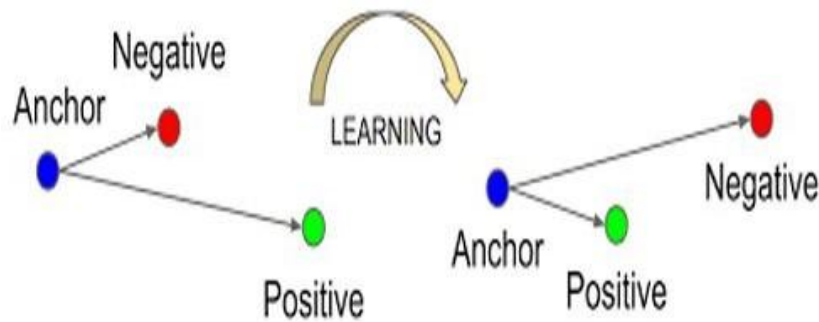


Fig 6.3.3 Functions of the system

White Box Testing:

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

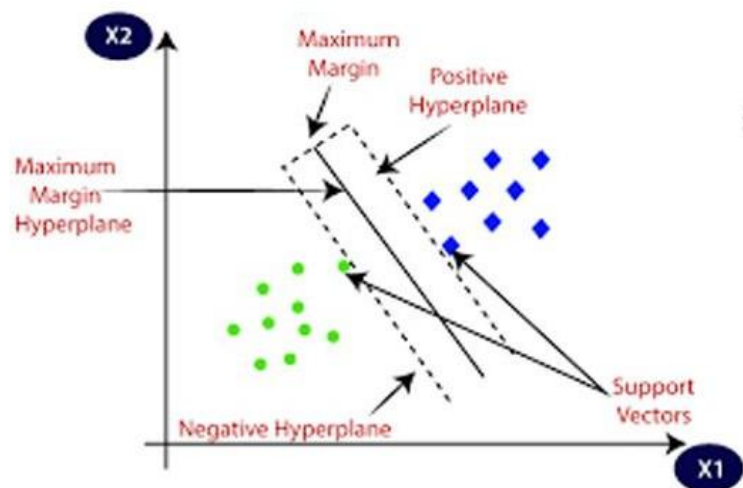


Fig 6.3.4 Tested features

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.

OUTPUT SCREENS

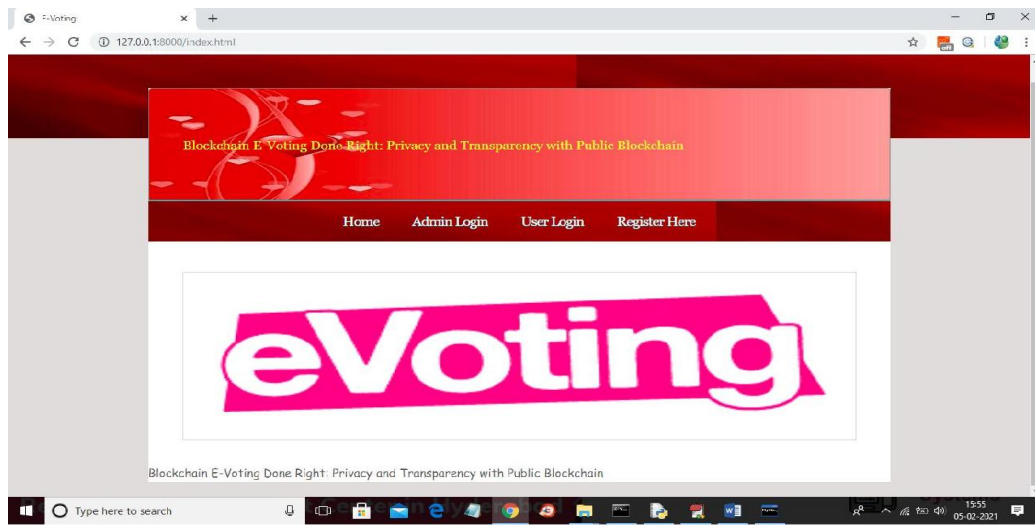


Fig.9.1: In above screen click on 'Admin Login' link to get below screen

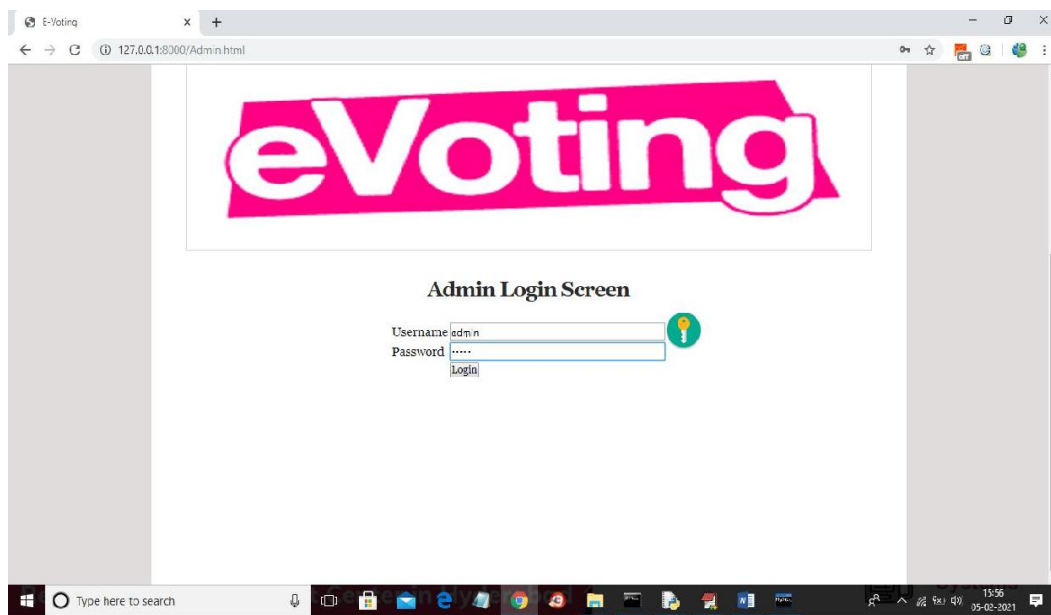


Fig.9.2: In above screen login as admin by giving username as 'admin' and password as 'admin' and then click Login button to get below screen

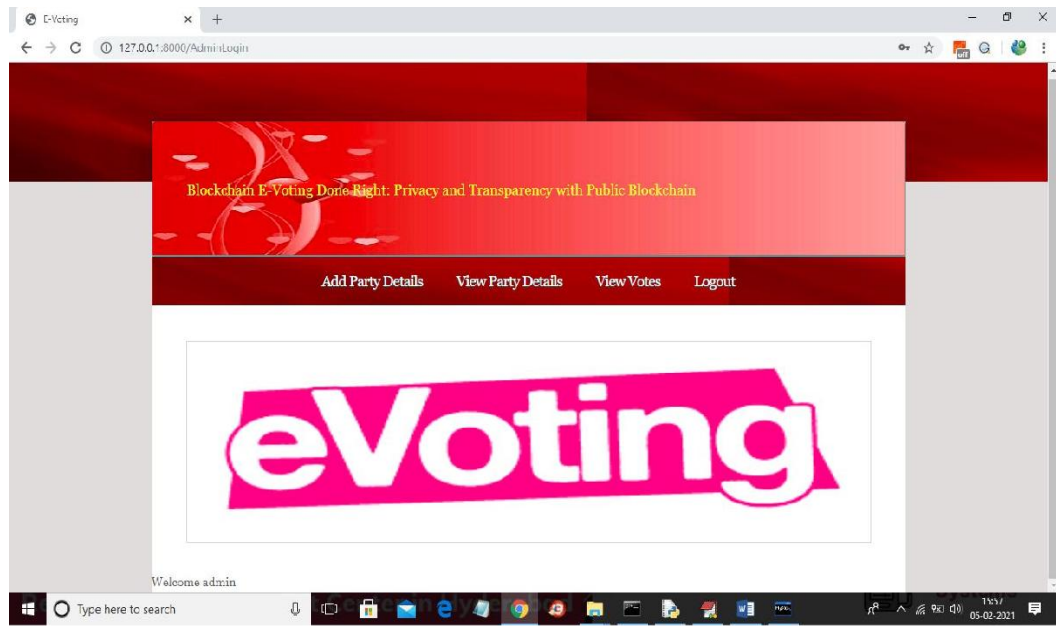


Fig.9.3: In above screen admin can click on ‘Add Party Details’ link to add party details

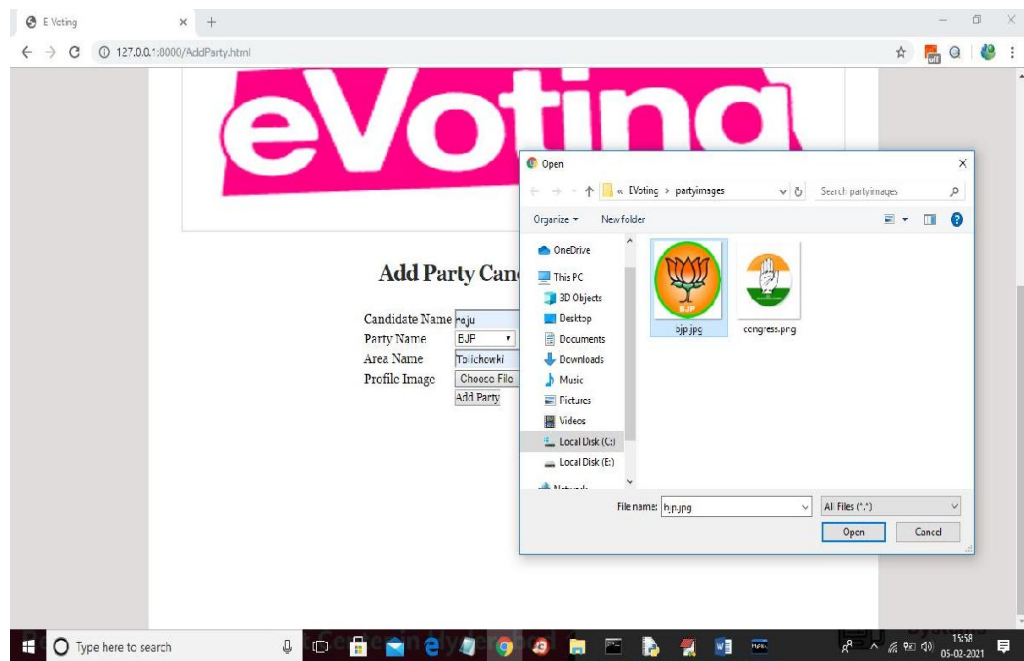


Fig.9.4: In above screen adding party and candidate details and then upload image and click on ‘Open’ button then click on ‘Add Party’ button to add party details

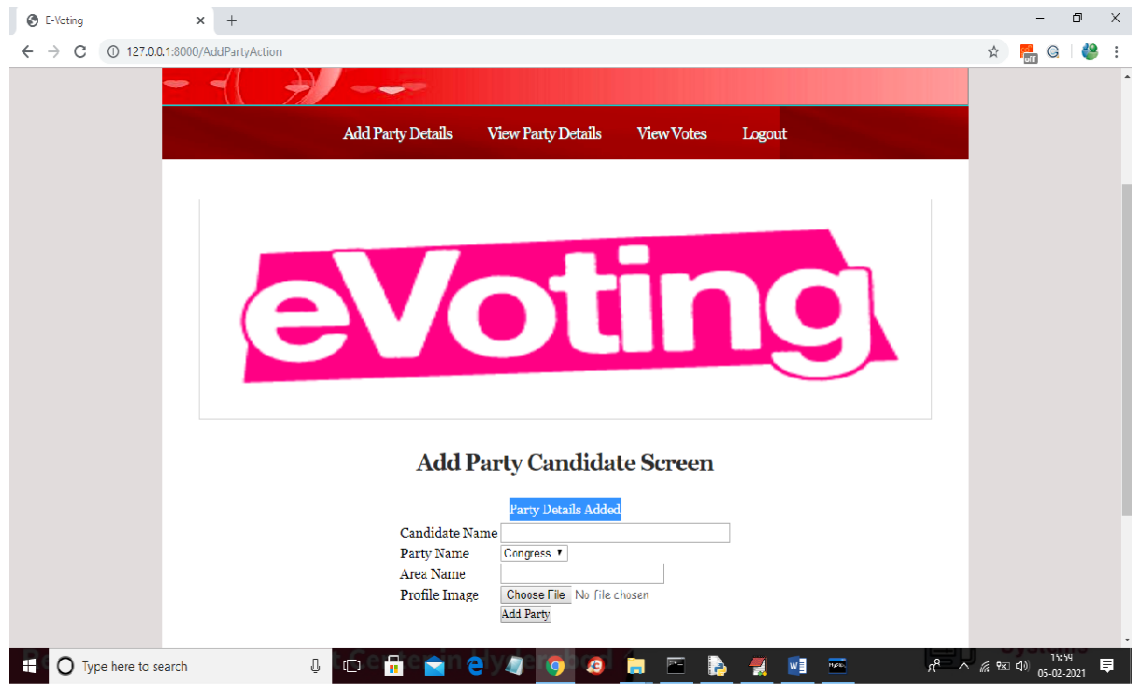


Fig.9.5: In above screen part details added and similarly you can add any number of party members and now click on ‘View Party Details’ link to get below screen

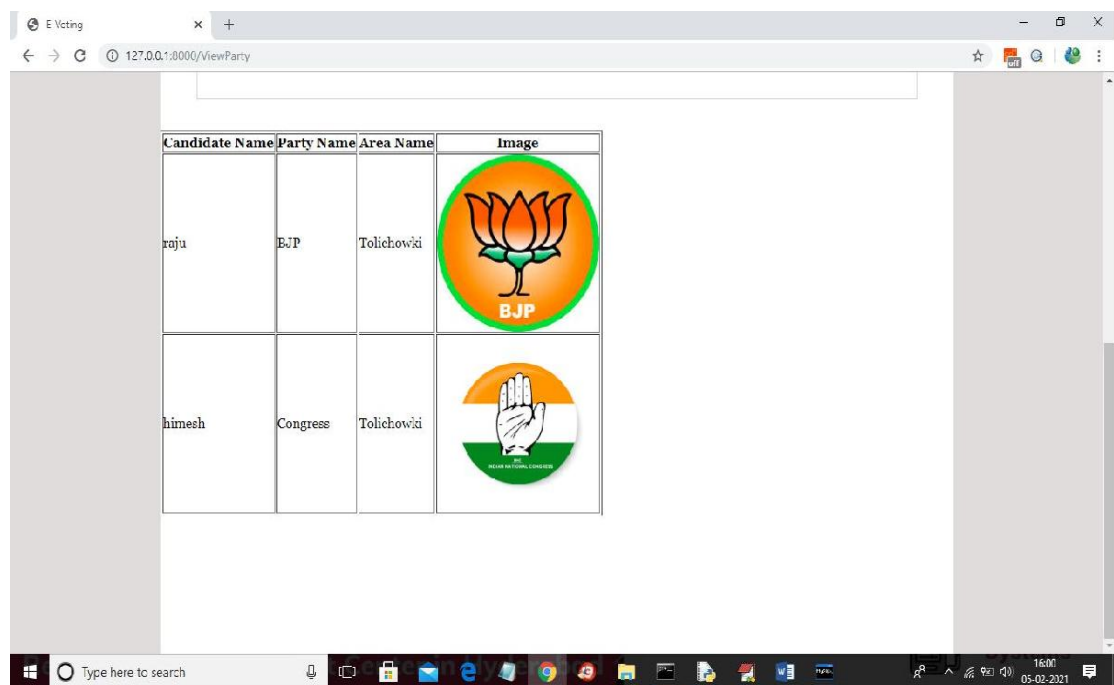


Fig.9.6: In above screen displaying add added party details and now click on ‘Logout’ link to logout as admin and then add new user

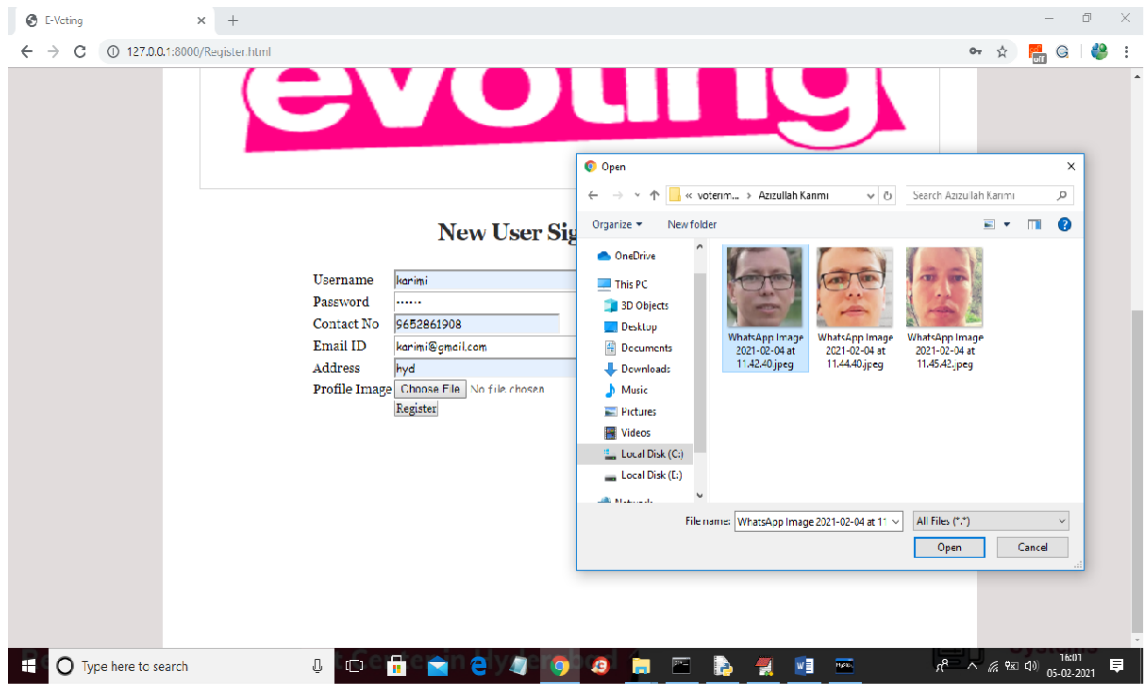


Fig.9.7: In above screen adding new user and then selecting his face photo taken from webcam and then click on ‘Register’ button to complete signup process. Here you have given images taken from phone but we need to capture from webcam for dataset as quality of webcam image and phone image vary and then problem comes in prediction.

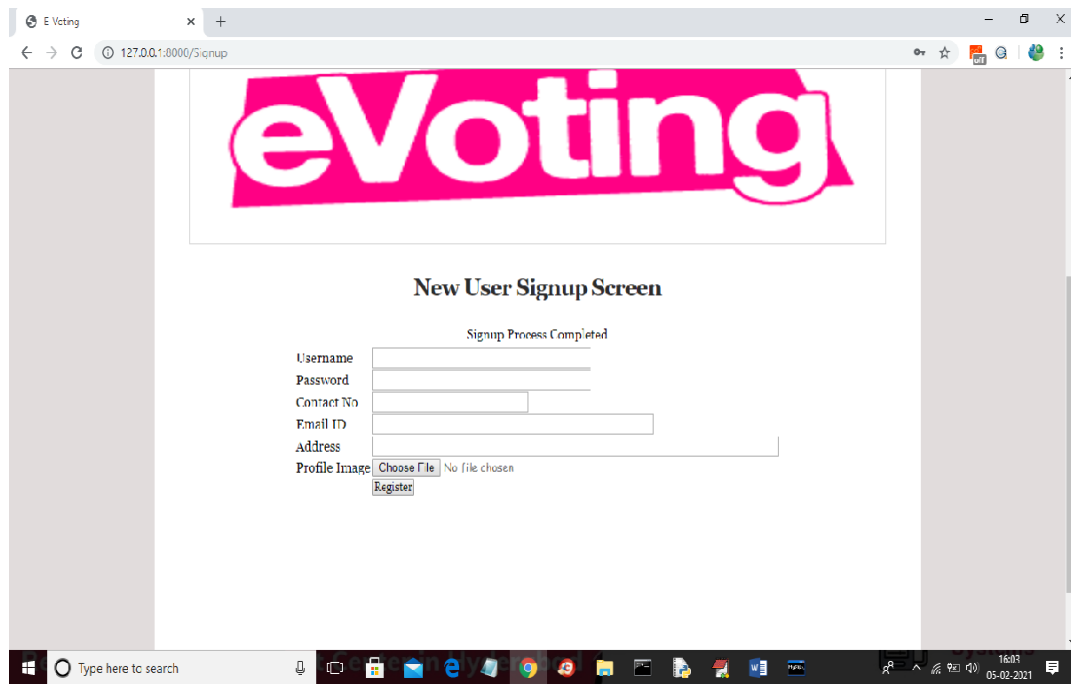


Fig.9.8: In above screen signup process completed and now login as this user to cast vote

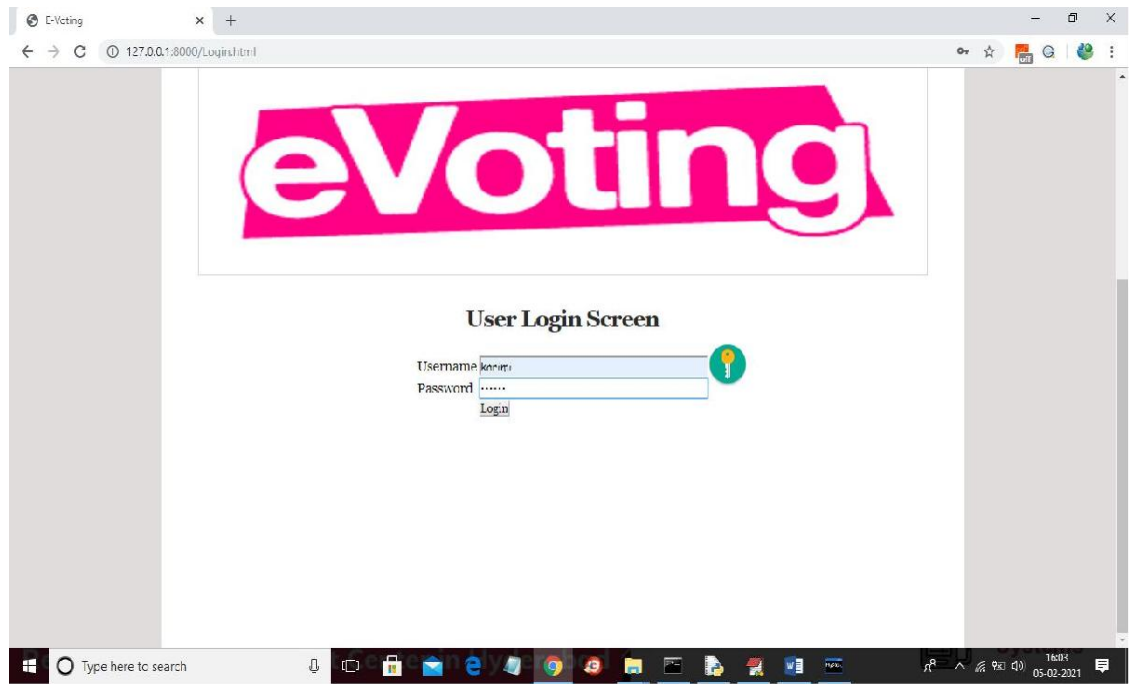


Fig.9.9: In above screen application first authenticate user by using his login details and once after successful login then user will get below page

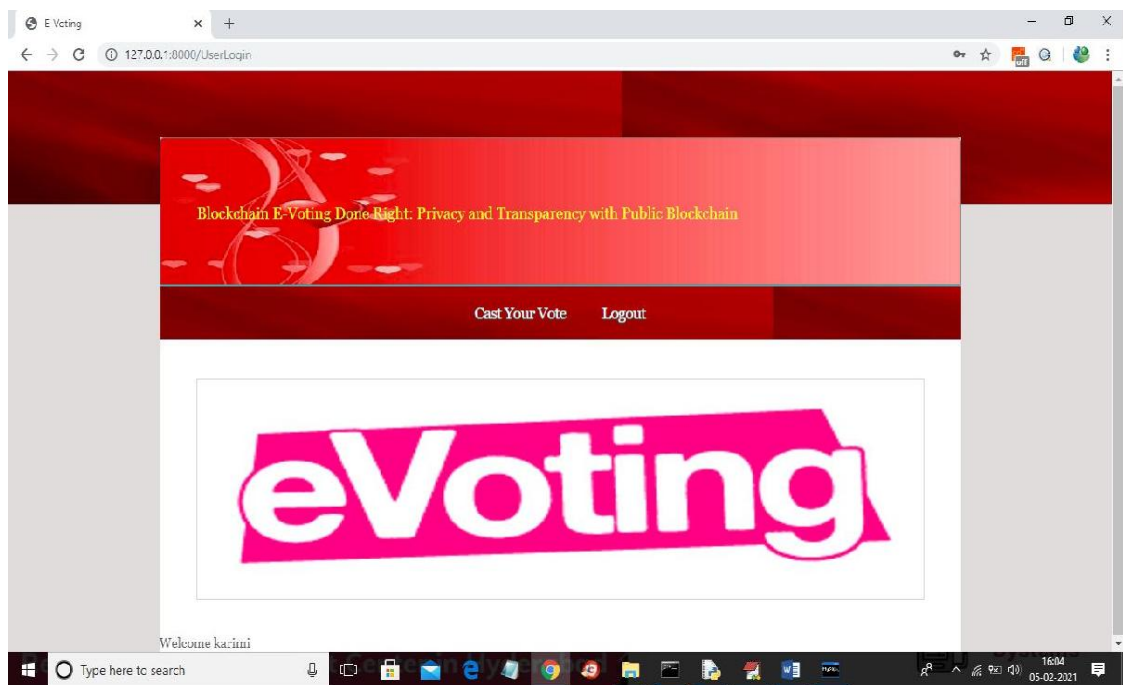


Fig.9.10: In above screen user can click on ‘Cast Your Vote’ link to get below webcam screen

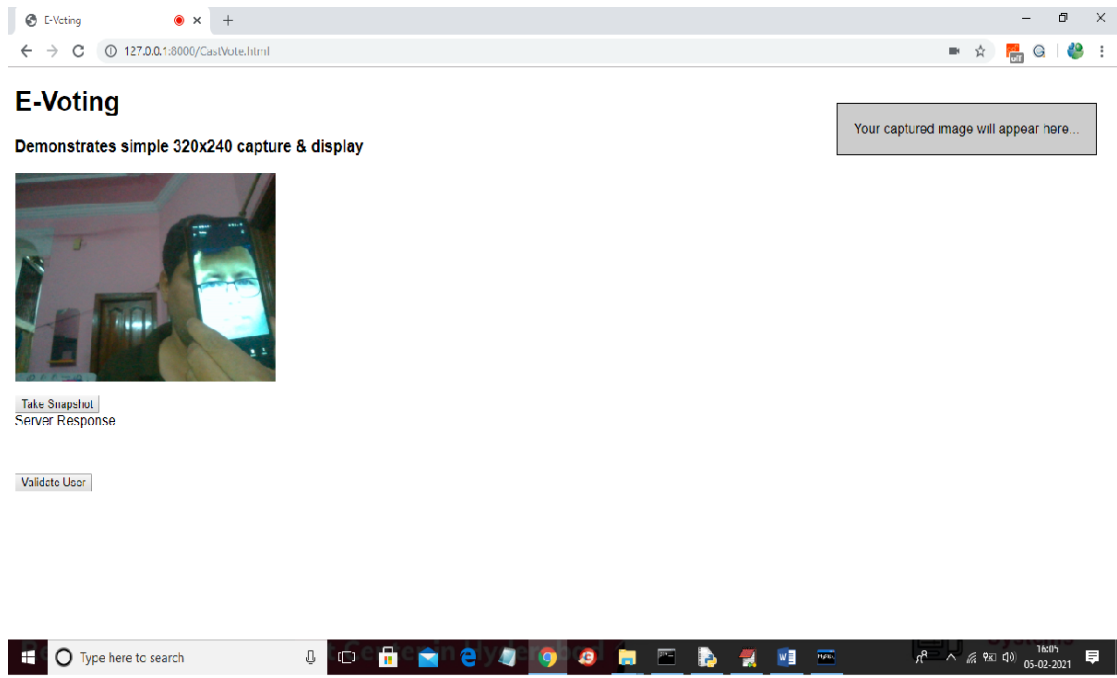


Fig.9.11: In above screen webcam is running and then by showing person face we need to click on ‘Take Snapshot’ button to capture his face

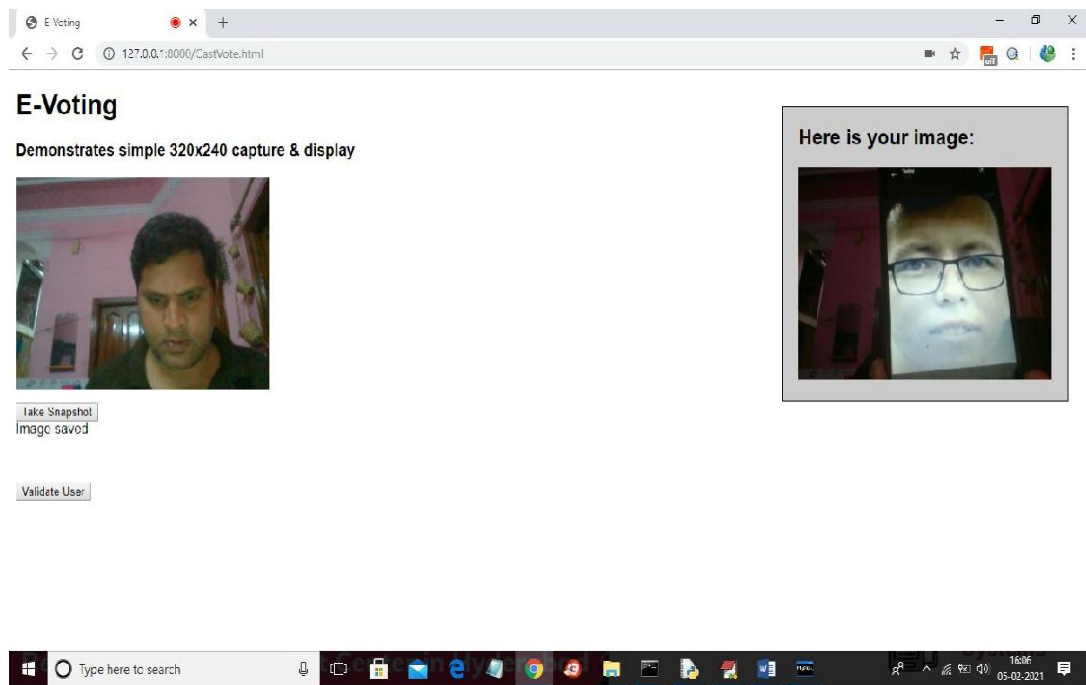


Fig.9.12: In above screen person face is capture and now click on ‘Validate User’ button to validate user

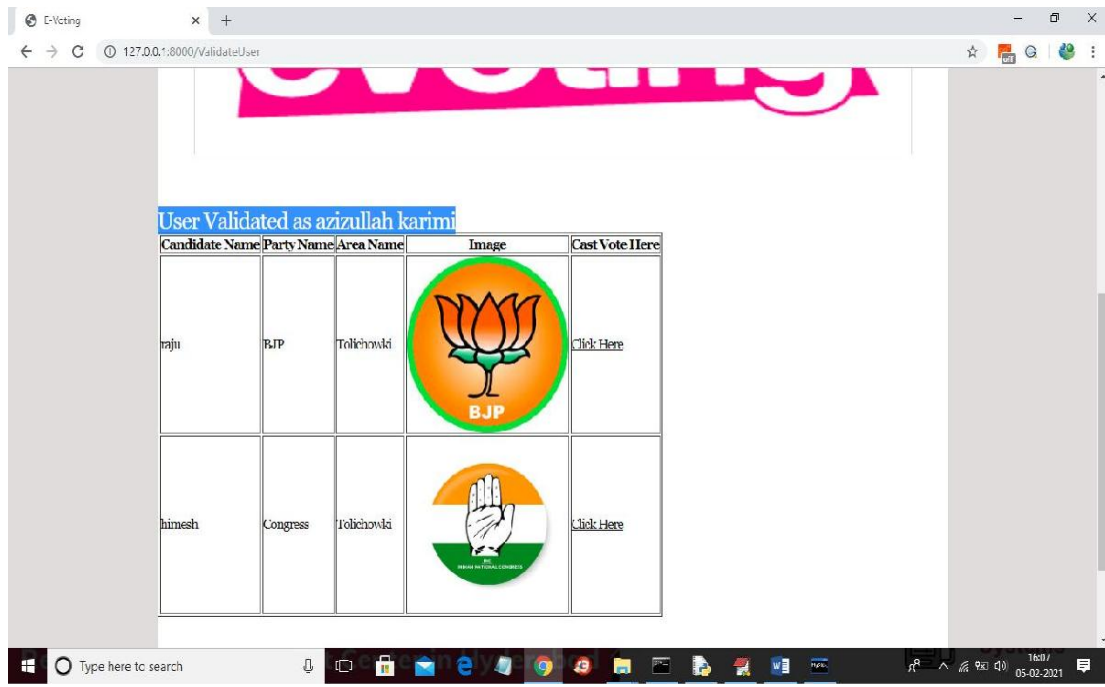


Fig.9.13: In above screen in blue colour you can see user is identified as ‘azizullahkarimi’ and then displaying list of candidates and now user can click on ‘Click Here’ option to cast his vote and to get below screen

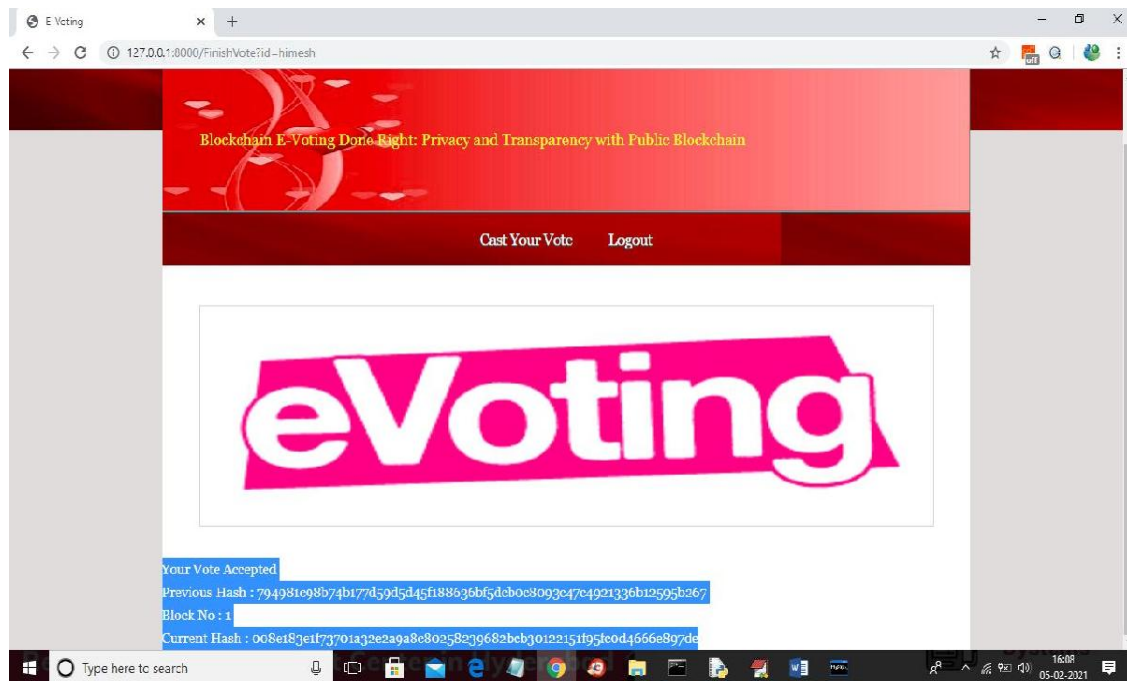


Fig.9.14: In above screen as this is the first vote so block will be added to Blockchain with block No as 1 and we can see Blockchain created a chain of blocks with previous and current hash code validation. Now try again with same user to cast vote

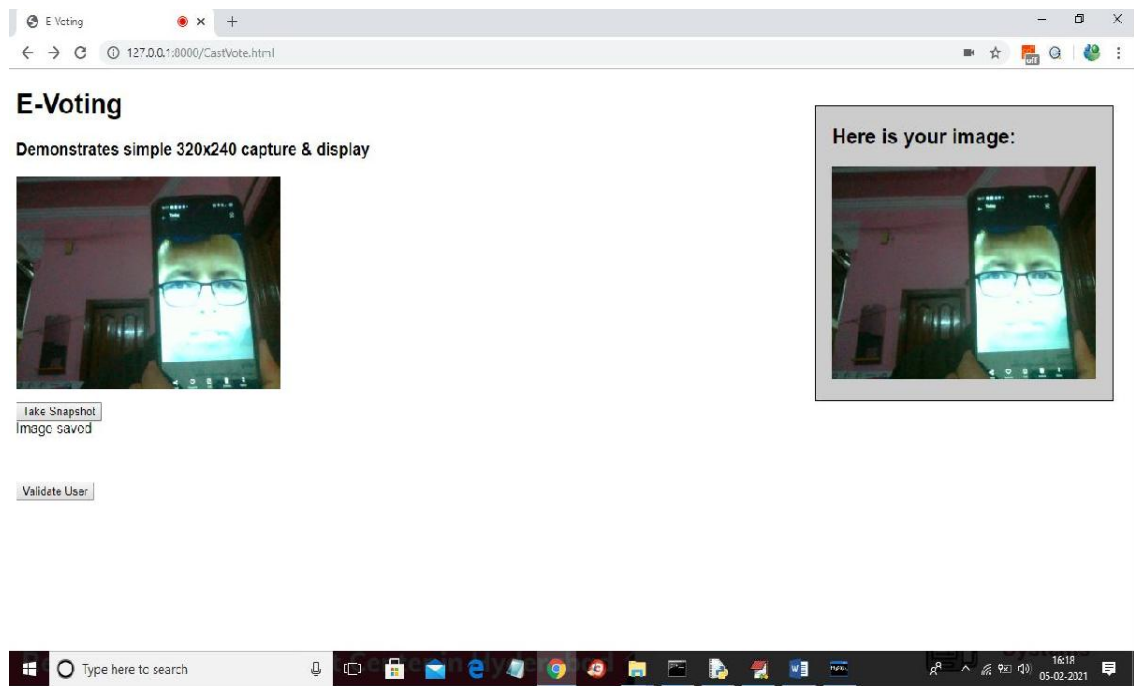


Fig.9.15: In above screen same user trying again and below is the result

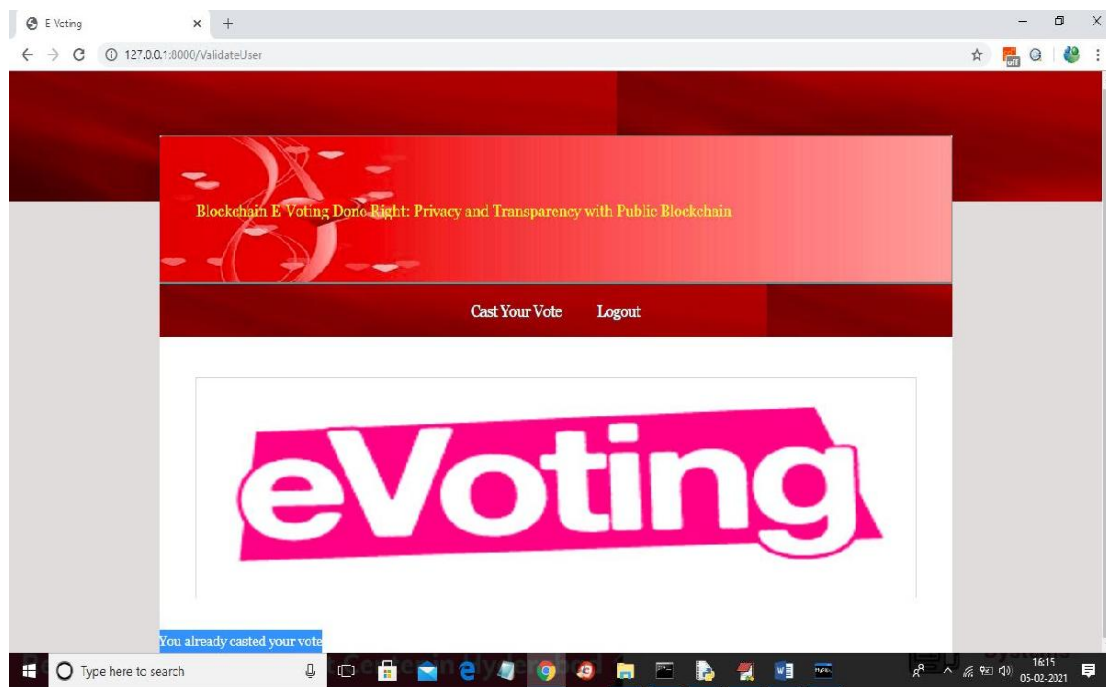


Fig.9.16: In above screen if same user try again then will get message as 'You already casted you vote' and now logout and login as 'admin' to get vote count

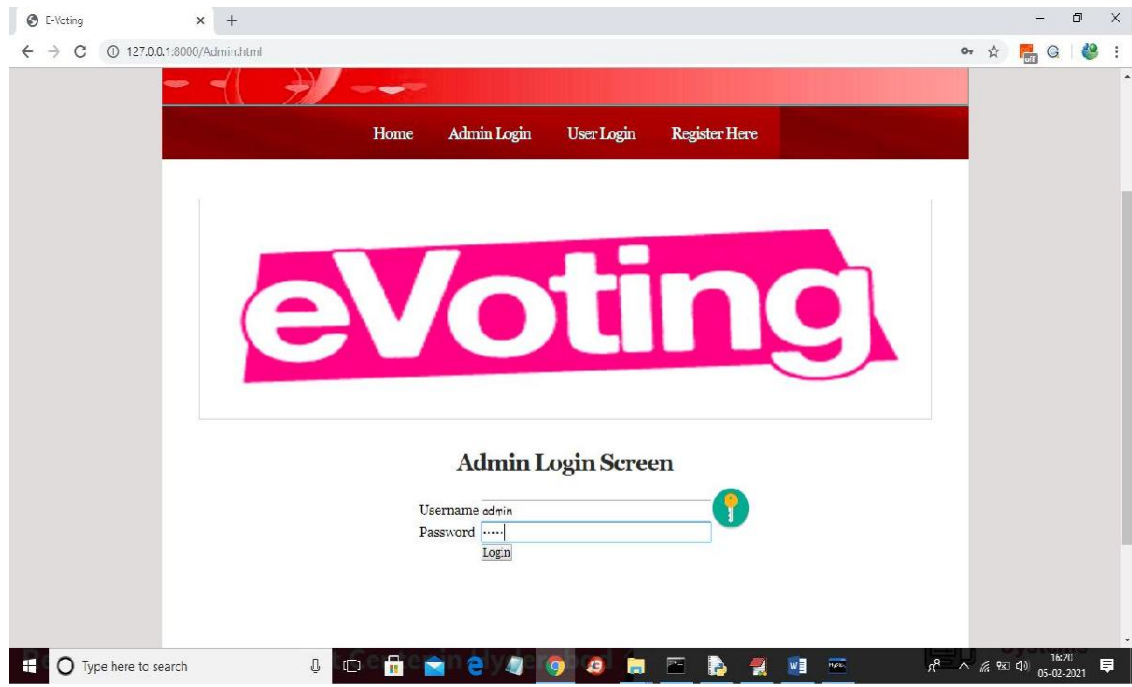


Fig.9.17: In above screen login as admin and after login will get below screen

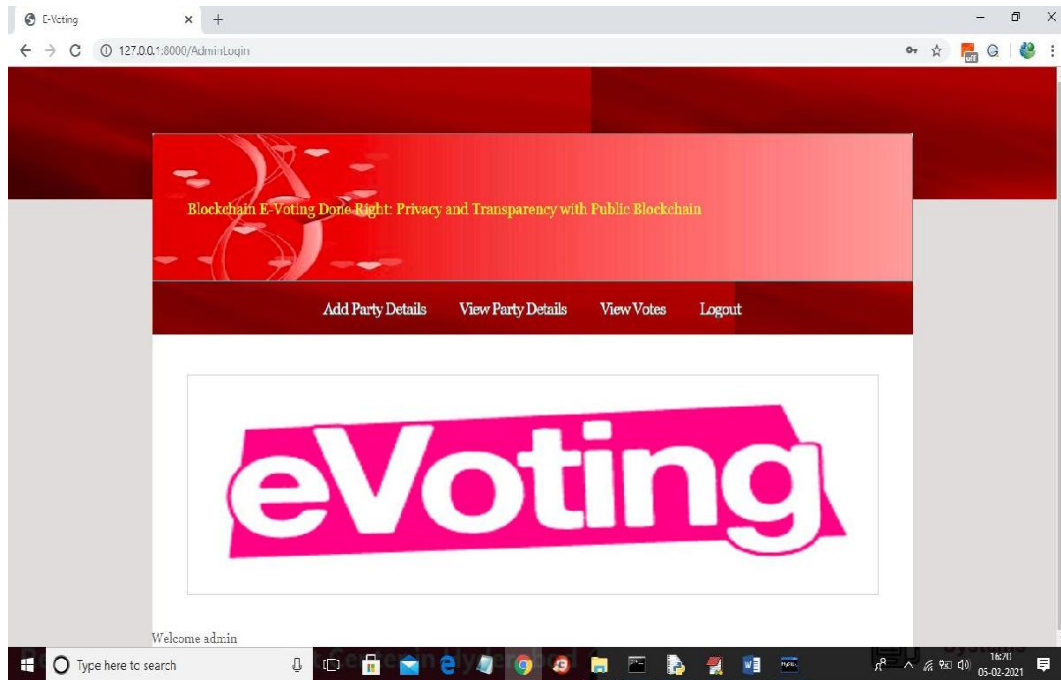


Fig.9.18: In above screen admin can click on 'View Votes' link to get below screen

The screenshot shows a web browser window titled 'E-Voting' with the address bar displaying '127.0.0.1:8000/ViewVotes'. The main content area contains a table with the following data:

Candidate Name	Party Name	Area Name	Image	Vote Count
raju	BJP	Tolichowki		0
himesh	Congress	Tolichowki		1

The Windows taskbar at the bottom shows the search bar, task view button, and various application icons. The system clock indicates the time is 16:27 on 05-02-2021.

Fig.9.19: In above screen admin can click on ‘Vote Count Display’ link to get the screen

FUTURE ENHANCEMENT

Future Enhancement is being planned to further analyze and enhance the protocol to a private blockchain is a bit faster, but it reduces the credibility of the whole system by being partially centralized because it only runs where the authority wants it. Future enhancements are being planned to upgrade the protocol to a private blockchain, which would increase processing speed, but at the cost of reduced credibility due to partial centralization. By only operating in areas authorized by the governing body, the system's decentralized nature would be compromised, potentially undermining trust in the network. This trade-off between speed and decentralization highlights the challenges of balancing efficiency with the core principles of blockchain technology, which emphasizes decentralization and transparency

CONCLUSION

Although we can see slight differences in network times, they are so negligible that public blockchain has more advantages in such an electoral system due to its openness of data and that anyone can watch them in the real time. A private blockchain is a bit faster, but it reduces the credibility of the whole system by being partially centralized because it only runs where the authority wants it. The table shows that the average times to add one person's voice are: Ganache 6.32 s (median 6.34 s), Hyperledger Composer 6.05 s (median 6.04 s), and Ethereum Ropsten 17.75 s (median 17.93 s). These times are influenced by the used consensus algorithm and also by the block time.

REFERENCE

- [1] N. Kshetri and J. Voas, "Blockchain-Enabled E-Voting," IEEE Software, vol. 35, pp. 95-99, jul 2018.
- [2] M. Pawlak, J. Guziur, and A. Poniszewska-Mara nda, "Voting Process with Blockchain Technology: Auditable Blockchain Voting System," in Lecture Notes on Data Engineering and Communications Technologies, pp. 233-244, Springer, Cham, 2019.
- [3] B. Singhal, G. Dhameja, and P. S. Panda, "How Blockchain Works," in Beginning Blockchain, pp. 31-148, Berkeley, CA: Apress, 2018.
- [4] Agora, "Agora Whitepaper," 2018.
- [5] R. Perper, "Sierra Leone is the first country to use blockchain during an election - Business Insider," 2018
- [6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," tech.rep., 2008.
- [7] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper, vol. 151, pp. 1-32, 2014.
- [8] S. Landers, "Netvote: A Decentralized Voting Platform - Netvote Project Medium," 2018.
- [9] P. McCorry, S. F. Shahandashti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," in Lecture Notes in Computer Science, ch. FCDS, pp. 357-375, Springer, Cham, 2017.
- [10] Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard)LWE," SIAM Journal on Computing, vol. 43, pp. 831-871, jan 2014

APPENDIX

```
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
import pymysql
from django.http import HttpResponse
from django.core.files.storage import FileSystemStorage
import os
from keras.utils.np_utils import to_categorical
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import model_from_json
from Blockchain import *
from Block import *
from datetime import date
import cv2
import numpy as np
import pyaes, pbkdf2, binascii, os, secrets
import base64
import random
import smtplib
from email.message import EmailMessage
from datetime import datetime

global load_model
load_model = 0
global classifier
```

```
global email_id
```

```
global otp
```

```
blockchain = Blockchain()
```

```
if os.path.exists('blockchain_contract.txt'):
```

```
    with open('blockchain_contract.txt', 'rb') as fileinput:
```

```
        blockchain = pickle.load(fileinput)
```

```
    fileinput.close()
```

```
names = ['Azizullah Karimi', 'Hasibullah Atayi', 'khushhal Qasimyar',
```

```
'Nazif Mal',
```

```
        'pacha khan khogyani', 'Ramin paikar', 'SayedAhmad Seyar sawiz',
```

```
'Venket Rao', 'wahidullah Bdri', 'yahya Maqsidi']
```

```
face_detection =
```

```
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
def getKey(): #generating key with PBKDF2 for AES
```

```
    password = "s3cr3t*c0d3"
```

```
    passwordSalt = '76895'
```

```
    key = pbkdf2.PBKDF2(password, passwordSalt).read(32)
```

```
    return key
```

```
def encrypt(plaintext): #AES data encryption
```

```
    aes = pyaes.AESModeOfOperationCTR(getKey(),
```

```
pyaes.Counter(31129547035000047302952433967654195398124239844
```

```
566322884172163637846056248223))
```

```
    ciphertext = aes.encrypt(plaintext)
```



```

    return ciphertext

def decrypt(enc): #AES data decryption
    aes = pyaes.AESModeOfOperationCTR(getKey(),
pyaes.Counter(31129547035000047302952433967654195398124239844
566322884172163637846056248223))
    decrypted = aes.decrypt(enc)
    return decrypted

def AddParty(request):
    if request.method == 'GET':
        return render(request, 'AddParty.html', { })

def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', { })

def Login(request):
    if request.method == 'GET':
        return render(request, 'Login.html', { })

def CastVote(request):
    if request.method == 'GET':
        return render(request, 'CastVote.html', { })

def Register(request):
    if request.method == 'GET':

```

```

        return render(request, 'Register.html', { })

def Admin(request):
    if request.method == 'GET':
        return render(request, 'Admin.html', { })

def getCurrentHour():
    now = datetime.now()
    dt = str(now)
    arr = dt.split(" ")
    arr = arr[1].strip().split(":")
    return int(arr[0])

def WebCam(request):
    if request.method == 'GET':
        data = str(request)
        formats, imgstr = data.split(';base64,')
        imgstr = imgstr[0:(len(imgstr)-2)]
        data = base64.b64decode(imgstr)
        with
open('C:/Users/prasa/OneDrive/Documents/EVoting/EVotingApp/static/p
hoto/test.png', 'wb') as f:
            f.write(data)
        f.close()
        context= {'data':"done"}
        return HttpResponse("Image saved")

def checkUser(name):

```

```

flag = 0
for i in range(len(blockchain.chain)):
    if i > 0:
        b = blockchain.chain[i]
        data = b.transactions[0]
        data = base64.b64decode(data)
        data = str(decrypt(data))
        data = data[2:len(data)-1]
        print(data)
        arr = data.split("#")
        if arr[0] == name:
            flag = 1
            break
return flag

```

```

def getOutput(status):
    output = '<h3><br/>'+status+'<br/><table border=1 align=center>'
    output+='<tr><th><font size=3 color=black>Candidate
Name</font></th>'
    output+='<th><font size=3 color=black>Party Name</font></th>'
    output+='<th><font size=3 color=black>Area Name</font></th>'
    output+='<th><font size=3 color=black>Image</font></th>'
    output+='<th><font size=3 color=black>Cast Vote
Here</font></th></tr>'
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = "", database = 'evoting',charset='utf8')
    with con:
        cur = con.cursor()
        cur.execute("select * FROM addparty")

```

```

rows = cur.fetchall()
for row in rows:
    cname = row[0]
    pname = str(row[1])
    area = row[2]
    image = row[3]
    output+='<tr><td><font size=3
color=black>'+cname+'</font></td>'
    output+='<td><font size=3 color=black>'+pname+'</font></td>'
    output+='<td><font size=3 color=black>'+area+'</font></td>'
    output+='<td><img src=/static/profiles/'+cname+'.png width=200
height=200></img></td>'
    output+='<td><a href=\'FinishVote?id='+cname+'\'><font size=3
color=black>Click Here</font></a></td></tr>'
    output+="</table><br/><br/><br/><br/><br/><br/>"
    return output

```

```

def FinishVote(request):
    if request.method == 'GET':
        cname = request.GET.get('id', False)
        voter = ""
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip("\n")
            file.close()
        today = date.today()
        data = str(user)+"#"+str(cname)+"#"+str(today)
        enc = encrypt(str(data))

```

```

enc = str(base64.b64encode(enc),'utf-8')
blockchain.add_new_transaction(enc)
hash = blockchain.mine()
b = blockchain.chain[len(blockchain.chain)-1]
print("Previous Hash : "+str(b.previous_hash)+" Block No :
"+str(b.index)+" Current Hash : "+str(b.hash))
bc = "Previous Hash : "+str(b.previous_hash)+"<br/>Block No :
"+str(b.index)+"<br/>Current Hash : "+str(b.hash)
blockchain.save_object(blockchain,'blockchain_contract.txt')
context= {'data': '<font size=3 color=black>Your Vote
Accepted<br/>'+bc}
return render(request, 'UserScreen.html', context)

```

```

def sendEmail():
    global email_id
    global otp
    msg = EmailMessage()
    msg.set_content("Your OTP is : "+str(otp))
    msg['Subject'] = 'E-Voting OTP'
    msg['From'] = "parreddynavyareddy@gmail.com"
    msg['To'] = "madasuannamani01@gmail.com"
    #print(email_id)

s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login("madasuprasannakumar38@gmail.com", "jrxxfhoqyynevdac")
s.send_message(msg)
s.quit()

```

```

def ValidateUser(request):
    if request.method == 'POST':
        global load_model
        global otp
        global classifier
        if
os.path.exists('C:/Users/prasa/OneDrive/Documents/EVoting/EVotingApp/static/photo/test.png') == True:

os.remove('C:/Users/prasa/OneDrive/Documents/EVoting/EVotingApp/static/photo/test.png')
        myfile = request.FILES['t1']
        fs = FileSystemStorage()
        filename =
fs.save('C:/Users/prasa/OneDrive/Documents/EVoting/EVotingApp/static/photo/test.png', myfile)
        user = "
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip("\n")
        file.close()
        option = 0
        status = "unable to predict user"
        img = cv2.imread('C:/Users/prasa/OneDrive/Documents/EVoting/EVotingApp/static/photo/test.png')
        faces = face_detection.detectMultiScale(img,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)
        print("===== "+str(len(faces)))

```

```

if len(faces) > 0:
    if load_model == 0:
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            classifier = model_from_json(loaded_model_json)
            classifier.load_weights("model/model_weights.h5")
            #classifier._make_predict_function()
            print(classifier.summary())
            load_model = 1

    faces = sorted(faces, reverse=True, key=lambda x: (x[2] - x[0]) *
(x[3] - x[1]))[0]
    (fX, fY, fW, fH) = faces
    roi = img[fY:fY + fH, fX:fX + fW]
    img = cv2.resize(roi, (64,64))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,64,64,3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    preds = classifier.predict(img)
    predict = np.argmax(preds)
    name = names[predict]
    #name = name.lower()
    print(str(name)+" "+str(user))
    flag = checkUser(user)
    flag = 0
    if flag == 0:
        if user in name:
            status = "User Validated as "+name

```

```

        option = 1
    else:
        status = "<h3><font size='3' color='black'>Unable to predict.
Please retry"
    else:
        status = "<font size='3' color='black'>You already casted your
vote"
    else:
        status = "<font size='3' color='black'>unable to detect face"

if option == 1:
    ""
    output = getOutput(status)
    context= {'data':output}
    return render(request, 'VotePage.html', context)
    ""
    otp = random.randint(1000,5000)
    sendEmail()
    print(otp)
    context= {'data':"OTP sent to your mail"}
    return render(request, 'OTPValidation.html', context)
else:
    context= {'data':status}
    return render(request, 'UserScreen.html', context)

```