

An Industrial Oriented Mini Project Report on

E – Voting System using Blockchain Technology (EtherVote)

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (AI & ML)

By

M. PRANAVA SESHASAI	21BD1A6657
G. PRANEETH SAI	21BD1A664J
G. MOUNEESH	21BD1A664H
P. SAI RAHUL	21BD1A665N

Under the guidance of

Mr. M. Srinivas

Assistant Professor, Department of CSE (AI & ML)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.

Narayanaguda , Hyderabad, Telangana-29

2024-25

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(An Autonomous Institute)



(Accredited by NBA & NAAC, Approved by A.I.C.T.E.)

Reg by Govt of Telangana State & Affiliated to JNTU, Hyderabad)



Department of Computer Science & Engineering (AI & ML)

CERTIFICATE

This is to certify that Industrial Oriented Mini Project work entitled “**E – Voting System using Blockchain Technology (EtherVote)**” is a bonafide work carried out in the seventh semester by

M. PRANAVA SESHASAI 21BD1A6657

G. PRANEETH SAI 21BD1A664J

G. MOUNEESH 21BD1A664H

P. SAI RAHUL 21BD1A665N

in partial fulfillment for the award of Bachelor of Technology in “**COMPUTER SCIENCE & ENGINEERING (AI&ML)**” from JNTU Hyderabad during the academic year 2024 - 2025 and no part of this work has been submitted earlier for the award of any degree.

INTERNAL GUIDE
(M. Srinivas)

HEAD OF THE DEPARTMENT
(Dr. T.V.G. Sridevi)

Submitted for Viva Voce Examination held on _____

External Examiner

Vision & Mission of KMIT

Vision of KMIT:

- To be the fountain head in producing highly skilled, globally competent engineers.
- Producing quality graduates trained in the latest software technologies and related tools and striving to make India a world leader in software products and services.

Mission of KMIT:

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish industry institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of The Department

Vision of The Department

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of The Department

- To provide faculty with state-of-the-art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES AND PO/PSO MAPPING

PROJECT OUTCOMES

P1: Secure online voting with facial recognition for fair elections.

P2: Transparent voting via blockchain to prevent tampering.

P3: Real-time election monitoring for accurate results.

P4: Inclusive remote voting access for all voters.

MAPPING PROJECT OUTCOMES WITH PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1			M	M	M		L	L	M	L	L	L
P2		M			L		M	L	M	L	M	M
P3				L	M	M	M	M	M	M	M	M
P4	M	M		M	M	M	M	M	M	L	M	M

L – Low

M – Medium

H – High

**PROJECT OUTCOMES MAPPING WITH
PROGRAM SPECIFIC OUTCOMES**

PSO	PSO1	PSO2
P1	M	L
P2	M	M
P3	M	M
P4	L	M

**PROJECT OUTCOMES MAPPING WITH PROGRAM
EDUCATIONAL OBJECTIVES**

PEO	PEO1	PEO2	PEO3	PEO4
P1		L	L	
P2		M	M	M
P3	M			M
P4	M		M	

DECLARATION

We hereby declare that the results embodied in the dissertation entitled “**E – Voting System using Blockchain Technology (EtherVote)**” has been carried out by us together during the academic year 2024-25 as a partial fulfillment of the award of the B.Tech degree in Computer Science and Engineering (AI & ML) from JNTUH. We have not submitted this report to any other university or organization for the award of any other degree.



STUDENT NAME

ROLL NO.

M. PRANAVA SESHASAI

21BD1A6657

G. PRANEETH SAI

21BD1A664J

G. MOUNEESH

21BD1A664H

P. SAI RAHUL

21BD1A665N

ACKNOWLEDGEMENTS

We take this opportunity to thank all the people who have rendered their full support to our project. We render our thanks to **Dr. B L Malleswari**, Principal, who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Founder & Director, **Mr. S. Nitin**, Director, for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Ms. Deepa Ganu**, Director Academics, for providing an excellent environment in the college.

We are also thankful to **Dr. T V G Sridevi**, Head of the Department, for providing us with time to make this project a success within the given schedule.

We are also thankful to our guide **Mr. M Srinivas**, Assistant Professor Department of CSE (AI&ML) for his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE (AI&ML) Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

STUDENT NAME

ROLL NO.

M. PRANAVA SESHSAI

21BD1A6657

G. PRANEETH SAI

21BD1A664J

G. MOUNEESH

21BD1A664H

P. SAI RAHUL

21BD1A665N

ABSTRACT

Democratic voting is a crucial and serious event in any place, the current election scheme in any place, be it a school college, or even a country is done through ballot papers or using EVM. This process has many disadvantages such as transparency, low voter turnout, vote tampering, lack of trust in electoral authorities, delay in results, and above all security issues. So the growing digital technology has helped many people's lives nowadays.

The concept of electronic voting is introduced to combat the disadvantages of the traditional voting system. Electronic voting is essentially an electronic means of casting and counting votes. It is an efficient and cost-effective way of conducting a voting procedure that is data- rich and real-time and requires high security. Nowadays, concerns about the security of networks and the privacy of communications for electronic voting have increased. Thus, the provision of electronic voting is very urgent and is becoming a popular topic in communication and networking.

One way to solve security problems is blockchain. The paper proposes a new blockchain-based electronic voting system that addresses some of the limitations in existing systems and evaluates some of the popular blockchain frameworks to create a blockchain-based electronic voting system. Because the blockchain stores its data in a decentralized manner, the implementation result shows that it is a practical and secure electronic voting system that solves the problem of vote forgery in electronic voting. The blockchain-based electronic voting system can be directly applied to various network applications.

LIST OF FIGURES

Figure 1 Blockchain Core Component	4
Figure 2 Class Diagram	11
Figure 3 Use Case Diagram.....	13
Figure 4 Sequence Diagram.....	15
Figure 5 DFD Level 0.....	16
Figure 6 DFD level 1	17
Figure 7 Event Trace Diagram	18
Figure 8 State Diagram.....	19
Figure 9 Overview Of System	22
Figure 10 Hash Table.....	24
Figure 11 Simplified Bitcoin Blockchain	25
Figure 12 Architecture of E-Voting.....	26
Figure 13 Smart Contract Working	29
Figure 14 Blockchain Transaction Flow.....	32
Figure 15 Find Face	33
Figure 16 Extract Facial Feature	34
Figure 17 Identify Faces	34
Figure 18 Project Timeline Chart	35
Figure 19 Admin Login	37
Figure 20 Admin Dashboard	38
Figure 21 View Users	38
Figure 22 Add User	39
Figure 23 Add Candidate.....	39
Figure 24 Add Election	40
Figure 25 Edit Phase.....	40
Figure 26 Elections.....	41
Figure 27 Candidates of Elections.....	41
Figure 28 User Login Page	42
Figure 29 Transaction Details.....	43
Figure 30 Result of Elections	44
Figure 31 Candidate Result	44

LIST OF TABLES

Table 1 Class Diagram Symbol Table	10
Table 2 Use Case Diagram Symbol Table.....	12
Table 3 Sequence Diagram Symbol Table	14
Table 4 Data Flow Diagram Symbol Table	16
Table 5 Event Trace Diagram Symbol Table	17
Table 6 State Diagram Symbol Table.....	19

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1 Aim of the project.....	1
1.2 Project Scope	1
1.3 Project Objectives	1
1.4 Project Modules	2
1.5 Project Basic Requirements	2
1.5.1 Hardware Requirements.....	2
1.5.2 Software Requirements	2
1.5.3 Software requirements for our clients	2
Chapter 2: Literature review.....	3
2.1 Introduction	3
Chapter 3: Analysis, Design Methodology and Implementation Strategy	8
3.1 Project Feasibility study	8
3.1.1 Economic Feasibility.....	8
3.1.2 Technical Feasibility	8
3.1.3 Operational Feasibility	8
3.2 Detailed Module Description.....	9
3.2.1 Login.....	9
3.2.2 Voting	9
3.2.3 Voting Creation.....	9
3.2.4 Voting Result	9
3.2.5 User Identification.....	9
3.3 Project SRS	10
3.3.1 Class Diagram.....	10
3.3.2 Use-case Diagram.....	11
3.3.3 Sequence Diagram	13
3.3.4 Data Flow Diagram	15
3.3.5 Event Trace Diagram	17
3.3.6 State Diagram	18

3.4 Methodology.....	20
3.4.1 Registration Stage.....	20
3.4.2 Login.....	20
3.4.3 Blockchain Technology.....	20
3.4.4 Ethereum.....	20
3.4.5 Database.....	21
3.4.6 Admin	21
3.4.7 Results Phase	21
3.4.8 Meta Mask.....	21
3.4.9 Truffle	21
3.5 Voting Process	22
3.6 Blockchain	23
3.6.1 Introduction.....	23
3.6.2 Solidity.....	28
3.6.3 Smart Contract	29
3.6.4 Ethereum.....	30
3.6.6 Truffle	31
3.6.7 Ganache	32
3.7 Face Recognition	33
3.7.1 Features.....	33
3.7.2 Command-Line Interface	35
3.8 Timeline Chart	35
Chapter 4: Implementation and Testing.....	36
4.1 Software and Tools	36
4.1.1 Hardware Requirements.....	36
4.1.2 Software Requirements	36
4.1.3 Software requirements for our clients	37
4.2 User Interface and Snapshots	37
4.2.1 Admin Login.....	37

4.2.2 Admin Dashboard	38
4.2.3 View Users	38
4.2.4 Add User	39
4.2.5 Add Candidate	39
4.2.6 Add Election	40
4.2.7 Edit Phase	40
4.2.8 View Elections	41
4.2.9 Candidate of Election	41
4.2.10 Login User	42
4.2.11 Metamask Transaction Details	43
4.2.12 Result of Elections	44
4.2.13 Result of Candidates	44
Chapter 5: Conclusion and Future work	45
5.1 Conclusion	45
5.2 Future work	45
Chapter 6: References	46

CHAPTER 1

1. INTRODUCTION

1.1 Aim of the project:

Online Voting system is a web-based voting system that will help to manage your elections easily and more securely. This voting system can be used for casting votes during the elections held in colleges, etc.

It uses face recognition technique to authenticate the user so problems for dummy entries will also be solved. So, thus this system gives the guarantee that no cheating can be done and the voting will be conducted easily where people don't have to go outside in order to cast their votes.

1.2 Project Scope:

Transparent transaction using Blockchain technology increases the security of the transaction of crypto that races towards the cybersecurity.

All the data is secure and verified. The encryption is done through cryptography to eliminate vulnerabilities such as unauthorized data tampering and this will increase the crypto payments.

1.3 Project Objectives:

- The main objective of the electronic voting technology intends to speed of the counting of ballots, reduce the cost of paying staff to count votes manually and can provide improved accessibility for disabled voters.
- This can be achieved by designing and developing a software platform for voter registration, election voting, real-time election results collation and monitoring and mostly for remote voters' access to elections.
- Study and implement a security method to be used to ensure that votes being cast in the system will not be compromised and no outside attack would be faced which will be ensured by blockchain technology.

1.4 Project Modules:

1. Login Module
2. Voting Module
3. Voting Creation Module
4. Voting Result Module
5. User Identification Module

1.5 Project Basic Requirements:

1.5.1 Hardware Requirements

- Any device containing camera with high processing power.

1.5.2 Software Requirements

- Front-End: React-js
- Back-End: Node-js, Express-js, Solidity
- Blockchain: MetaMask, Ganache
- Others: Domain name, Hosting, sever and cloud storage

1.5.3 Software requirements for our clients

- Windows 7 or higher OS
- Google chrome or any other safe browser
- Good Internet Speed
- Blockchain ID with sufficient Ethereum

CHAPTER 2

2. LITERATURE SURVEY

2.1 Introduction

Blockchain technology has several uses in various industries, such as banking and finance, IoT and media, energy, health, logistics, and many more. In addition, new fields and ways of use are continually being investigated. The authors of comprehensively evaluate several articles on Blockchain technology in information systems and give a comprehensive list of applications organized by Blockchain technology problems. This section provides a fundamental analysis of previous research that looked at scalable Blockchain technology for electronic voting. This research used digital libraries such as IEEE Xplore Digital Library, Scopus, ScienceDirect, SpringerLink, and ACM Digital Library to identify and review the existing literature to conduct a methodical study of such efforts. It allowed us to perform a rigorous study analysis of the efforts made. These are broad studies of Blockchain technology, and they mention scalability as a significant challenge for Blockchain. These examinations do not specifically investigate or address the scalability of Blockchain technology, and as a result, they have been excluded from the scope of the present study.

- Core components of blockchain architecture

1. Node — user or computer within the blockchain
2. Transaction — smallest building block of a blockchain system
3. Block — a data structure used for keeping a set of transactions which is distributed to all nodes in the network
4. Chain — a sequence of blocks in a specific order
5. Miners — specific nodes which perform the block verification process
6. Consensus— a set of rules and arrangements to carry out blockchain operations

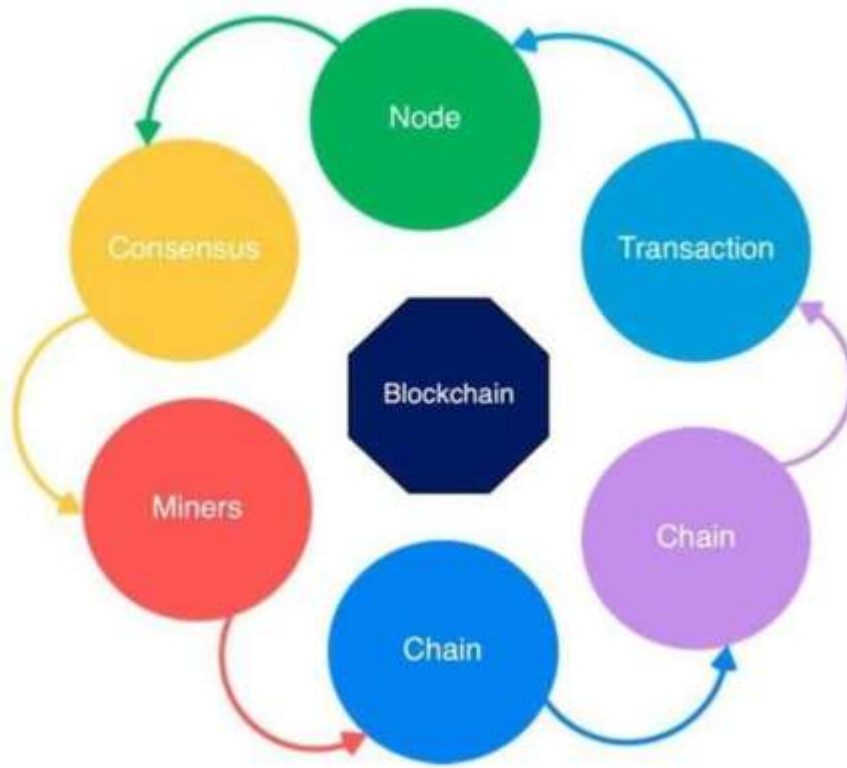


Figure 1: Blockchain Core Component

Currently, research is being carried out on using Blockchain technology in online voting systems. The authors of describe and compare numerous Blockchain-based electronic voting systems in their summary on voting. The deployment of Blockchain-based electronic voting systems presents both threats and potential.

Another significant survey was performed in, whose authors researched several Blockchain-based electronic voting systems to see how they adhere to existing international standards and conventions. To conclude, specific commercial Blockchain-based electronic voting systems are given and discussed in in addition to an open electronic voting platform.

The authors of investigated electronic voting obstacles and the present state of the Blockchain-based e- voting systems. According to, electronic voting may be unavoidable when comparing conventional voting with distant voting through the Internet.

Finally, the authors of present an overview of the current voting systems by describing several voting techniques, their benefits and drawbacks, and technical breakthroughs in the sector.

Numerous articles have been written on merging Blockchain technology with electronic voting systems. The results show how each method has distinct aims and is executed differently in these articles. These publications cover a variety of techniques to accomplish various objectives using a variety of ways. That is why we are performing an SLR on scalable Blockchain-based electronic voting to obtain a handle on what is new in the industry.

In conclusion, these studies lack both a systematic strategy for conducting the review and sufficient depth in current initiatives to address the scalability of Blockchain for electronic voting. Consequently, there are gaps in the coverage of several aspects of Blockchain scalability and the survey's depth and breadth. Furthermore, due to the ever-increasing length of the Bitcoin chain and recent advancements such as Segregated Witness (SegWit), the scalability of Blockchain has recently garnered significant interest.

Consequently, a current and extensive effort is necessary to systematically assess the scalability of Blockchain technology for electronic voting based on its present level of development. This evaluation aims to identify the work that has already been carried out, the existing limitations, and the potential avenues for future research.

The paper is based on using a technique called “systematic mapping”. It is organized around five research questions whose answers, according to the authors, cover the entire scope of their review:

(1) What are the current e-voting system gaps? (2) Can the blockchain concept improve e-voting systems? (3) What are the research topics and proposed solutions that have been published in blockchain-based e-voting? (4) Which blockchain platforms/consensus models are used? (5) What are the future research directions for the blockchain-based e-voting system? They have answered all these questions and more specifically question (3), which focuses on comparing between e-voting applications that are based on blockchain technology.

They classified these applications into 5 categories, where each category represents the main purpose of the application or its main feature. This classification is not enough and it is not based on an obvious comparison between the applications. In our paper in hand, we extracted several properties shared by most blockchain-based e-voting applications, and then we defined a subset of comparison criteria by grouping them (the extracted properties) into 4 themes and hence 4 tables. Primarily based on deep convolutional network techniques, Zhang adopted 3 ranges of deep

convolutional networks that may predict the coarse-to-exceptional position of face and landmarks superbly. A current look at has proven that during this discipline, deep getting to know methods can have vast effects.

The writer has advised CNNs for eye detection consisting of trio tiers: idea network (p-net), refinement community (r-net), and output network (o-net). Experimental consequences unearth these strategies to exceed trendy techniques over multiple disturbing assessments whilst keeping efficiency in actual-time.

Lai et al. suggested a decentralized anonymous transparent electronic voting system (DATE) requiring a minimal degree of confidence between participants. They think that for large-scale electronic elections, the current DATE voting method is appropriate. Unfortunately, their proposed system is not strong enough to secure from DoS attacks because there was no third-party authority on the scheme responsible for auditing the vote after the election process. This system is suitable only for small scales because of the limitation of the platform. Although using Ring Signature keeps the privacy of individual voters, it is hard to manage and coordinate several signer entities. They also use PoW consensus, which has significant drawbacks such as energy consumption: the “supercomputers” of miners monitor a million computations a second, which is happening worldwide. Because this arrangement requires high computational power, it is expensive and energy-consuming.

Shahzad et al. proposed the BSJC proof of completeness as a reliable electronic voting method. They used a process model to describe the whole system’s structure. On a smaller scale, it also attempted to address anonymity, privacy, and security problems in the election. However, many additional problems have been highlighted. The proof of labour, for example, is a mathematically vast and challenging job that requires a tremendous amount of energy to complete. Another problem is the participation of a third party since there is a significant risk of data tampering, leakage, and unfair tabulated results, all of which may impact end-to- end verification. On a large scale, generating and sealing the block may cause the polling process to be delayed.

Gao et al. has suggested a blockchain-based anti-quantum electronic voting protocol with an audit function. They have also made modifications to the code-based Niederreiter algorithm to make it more resistant to quantum assaults. The Key Generation Centre (KGC) is a certificateless cryptosystem that serves as a regulator. It not only recognizes the voter’s anonymity but also

facilitates the audit's functioning. However, an examination of their system reveals that, even if the number of voters is modest, the security and efficiency benefits are substantial for a small-scale election. If the number is high, some of the efficiency is reduced to provide better security.

Yi presented the blockchain-based electronic voting Scheme (BES) that offered methods for improving electronic voting security in the peer-to-peer network using blockchain technology. A BES is based on the distributed ledger (DLT) may be employed to avoid vote falsification. The system was tested and designed on Linux systems in a P2P network. In this technique, counter-measurement assaults constitute a significant issue. This method necessitates the involvement of responsible third parties and is not well suited to centralized usage in a system with many agents. A distributed process, i.e., the utilization of secure multipart computers, may address the problem. However, in this situation, computing expenses are more significant and maybe prohibitive if the calculation function is complex and there are too many participants.

Khan, K.M. has proposed block-based e-voting architecture (BEA) that conducted strict experimentation with permissioned and permissionless blockchain architectures through different scenarios involving voting population, block size, block generation rate, and block transaction speed. Their experiments also uncovered fascinating findings of how these parameters influence the overall scalability and reliability of the electronic voting model, including interchanges between different parameters and protection and performance measures inside the organization alone. In their scheme, the electoral process requires the generation of voter addresses and candidate addresses. These addresses are then used to cast votes from voters to candidates. The mining group updates the ledger of the main blockchain to keep track of votes cast and the status of the vote. The voting status remains unconfirmed until a miner updates the main ledger. The vote is then cast using the voting machine at the polling station.

However, in this model, there are some flaws found. There is no regulatory authority to restrict invalid voters from casting a vote, and it is not secure from quantum attack. Their model is not accurate and did not care about voter's integrity. Moreover, their scheme using Distributed consensus in which testimonies (data and facts) can be organized into cartels because fewer people keep the network active, a "51%" attack becomes easier to organize. This attack is potentially more concentrated and did not discuss scalability and delays in electronic voting, which are the main concerns about the blockchain voting system.

CHAPTER 3

3. ANALYSIS, DESIGN METHODOLOGY AND STRATEGY

3.1 Project Feasibility Study

Feasibility Study is the Measure of how beneficial or practical the development of the System will be to the Organization. It is a Preliminary survey for the Systems Investigation. It aims to provide information to facilitate a later in-depth investigation. The Report is Produced at the end of the feasibility study contains suggestions and reasoned arguments to help management decide whether to commit further resources to the proposed project.

3.1.1 Economic Feasibility

In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system will be Economically Feasible if it Successfully Developed with at least 95% Accuracy.

3.1.2 Technical Feasibility

This is the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platform.

3.1.3 Operational Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

3.2 Detailed Module Description

3.1.1 Login

This module describes about login functionality in which there are two types of logins are there 1) User Login 2) Admin Login.

3.1.2 Voting

In this module user can cast their votes. This module will describe a whole page where there will be list of candidates were noted with their symbols.

3.1.3 Voting Creation

In admin can create whole voting environment (i.e., arrange voting facility, specifies time of voting, add candidates that have stand, and records the entry that particular person has voted, etc).

3.1.4 Voting Result

This module is the main module of our system as it gives the result of the voting. As we have used blockchain technology our votes will not be not be altered and thus purity maintains and the proper candidate will be won.

3.1.5 User Identification

This module system will authenticate user that particular user is valid or not. This authentication will be done by face recognition system where user will have to scan their face to access the voting portal in order to make voting system crystal clear because due to this no other person can cast vote in place of a particular user.

3.2 Project SRS

3.2.1 Class Diagram

A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics.

3.2.1.1 Symbols

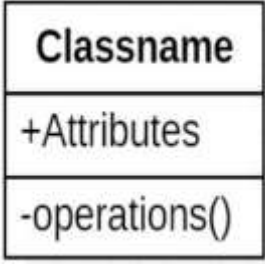
	<p>Classname: The name of the class is the only required tag in the graphical representation of a class. It always appears in the top-most compartment.</p> <p>Attributes: An attribute is a named property of a class that describes the object being modelled. In the class diagram, attributes appear in the second compartment just below the name-compartment.</p> <p>Attributes can</p> <p>be: + public #</p> <p>Protected</p> <p>- private</p> <p>/ Derived</p> <p>Operation: Operations describe the class behaviour and appear in the third compartment.</p>
-----------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 1 Class Diagram Symbol Table

3.2.1.2 Diagram

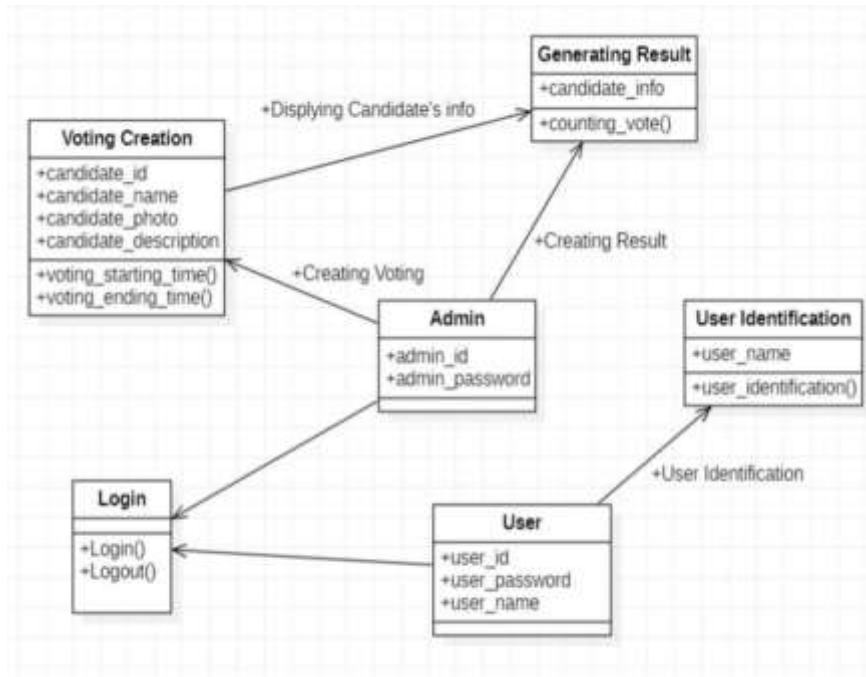


Figure 2 Class Diagram

3.2.2 Use-case Diagram

Use case diagrams are a common way to communicate the major functions of a software system. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Use cases are nothing but the system functionalities written in an organized manner. Now another thing which is relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

So, in brief, the purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

• Symbols



 Actor	An actor is rendered as a stick figure in a use case diagram. Each actor participates in one or more use cases.
 Use Case	People, processes, or other systems that interact with information systems that will be created outside the information system that will be created themselves.

Table 2 Use Case Diagram Symbol Table

• Diagram

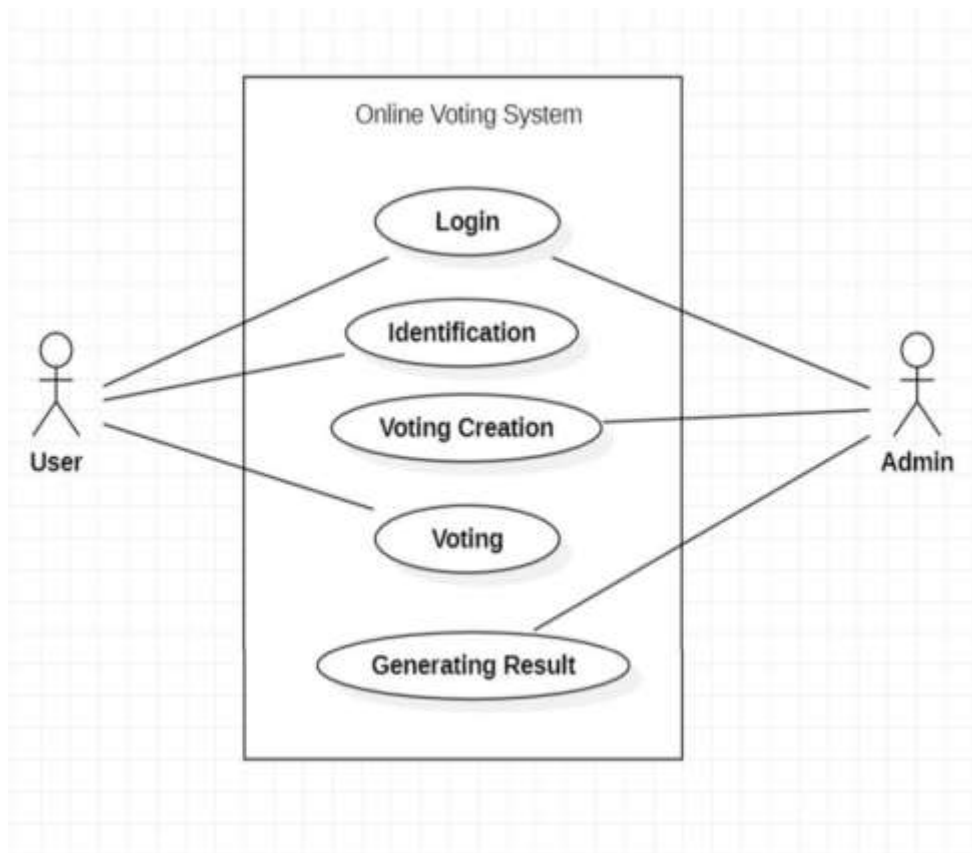


Figure 3 Use Case Diagram

3.2.3 Sequence Diagram

Sequence diagram represents the behavioral aspects of a system. Sequence diagram shows the interactions between the objects by means of passing messages from one object to another with respect to time in a system.

Sequence diagram contains the objects of a system and their life-line bar and the messages passing between them. Objects appear at the top portion of sequence diagram. Object is shown in a rectangle box. Name of object precedes a colon ':' and the class name, from which the object is instantiated. The whole string is underlined and appears in a rectangle box. A down-ward vertical line from object-box is shown as the life- line of the object. A rectangle bar on life-line indicates that it is active at that point of time. Messages are shown as an arrow from the life-line of sender object to the life-line of receiver object and labelled with the message name.

3.2.3.1 Symbols

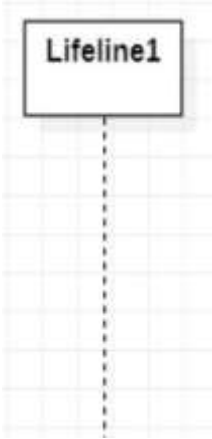
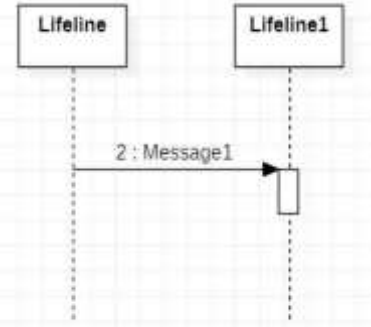
	<p>A Lifeline in a sequence diagram is rendered as a box with a dashed line descending from it. The line is called the object lifeline, and it represents the existence of an object over a period of time.</p>
	<p>Messages are rendered as horizontal arrows being passed from object to object as time advances down the object lifelines. Conditions indicate when a message gets passed.</p>

Table 3 Sequence Diagram Symbol Table

3.2.3.2 Diagram

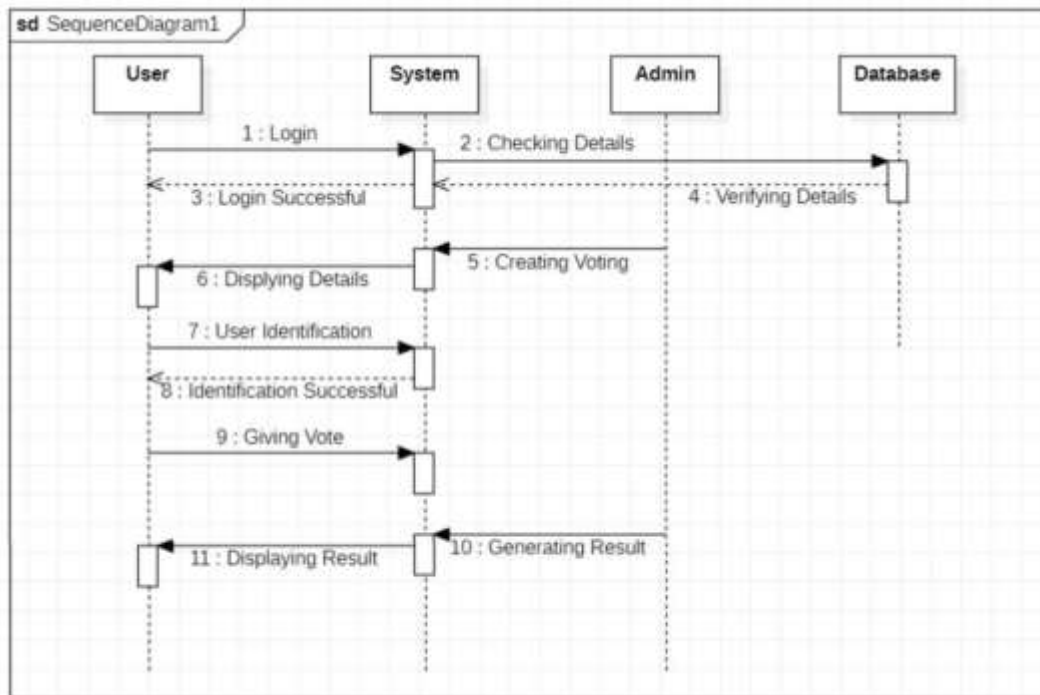


Figure 4 Sequence Diagram

3.2.4 Data Flow Diagram

DFD provides the functional overview of a system. The graphical representation easily overcomes any gap between 'user and system analyst' and 'analyst and system designer' in understanding a system. Starting from an overview of the system it explores detailed design of a system through a hierarchy.

DFD shows the external entities from which data flows into the process and also the other flows of data within a system. It also includes the transformations of data flow by the process and the data stores to read or write a data.

3.2.4.1 Symbols

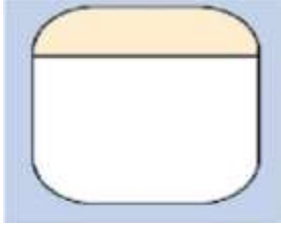
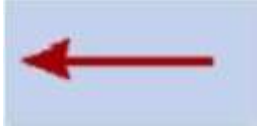


	Process
	Data Flow
	Data Store
	Source / Sink

Table 4 Data Flow Diagram Symbol Table

3.2.4.2 Diagram

3.2.4.3 DFD level 0:

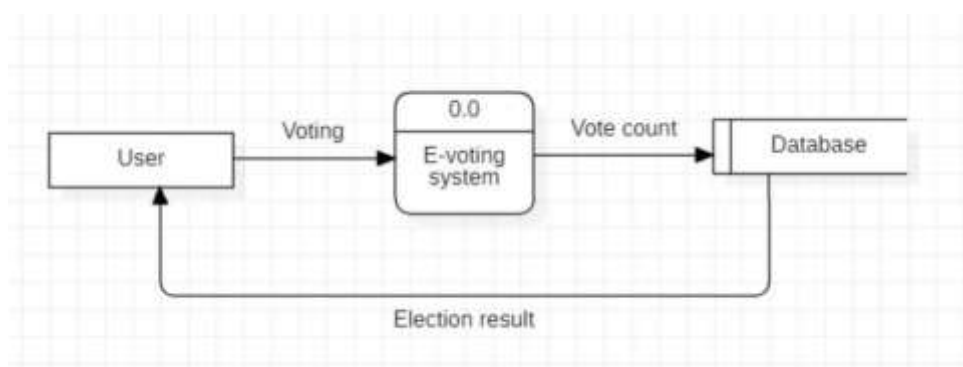


Figure 5 DFD Level 0

3.2.4.4 DFD level 1:

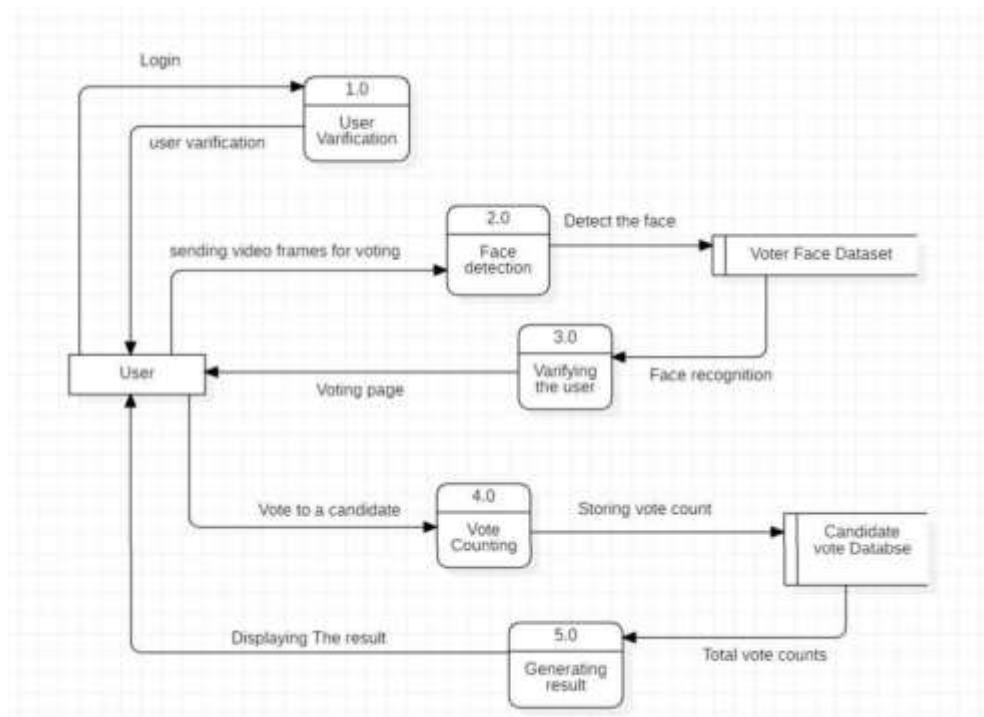


Figure 6 DFD level 1

3.2.5 Event Trace Diagram

An event Trace diagram simply depicts interaction between objects in a sequential order. i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.

3.2.5.1 Symbols




	Start State
	Final State
	Simple State

Table 5 Event Trace Diagram Symbol Table

3.2.5.2 Diagram

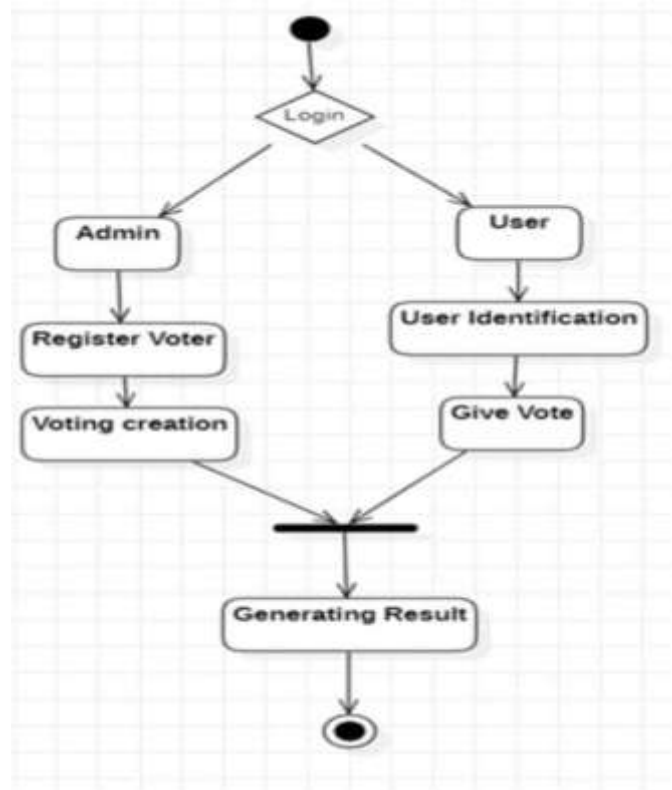


Figure 7 Event Trace Diagram

3.2.6 State Diagram

A state diagram is a type of diagrams used in computer Science and related fields to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

3.2.6.1 Symbols





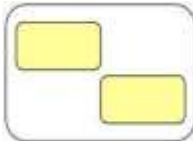
	Start State
	Final State
	Simple State
	Concurrent Composite State
	Sequential Composite State

Table 6 State Diagram Symbol Table

3.2.6.2

3.2.6.3 Diagram

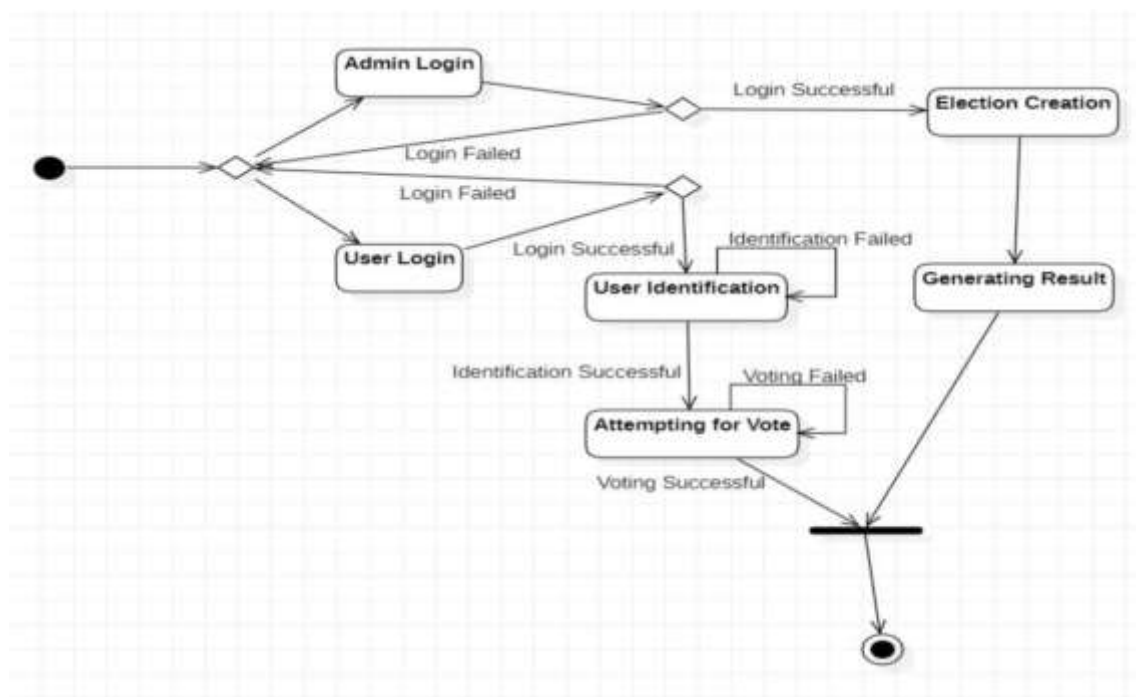


Figure 8 State Diagram

3.3 Methodology

In this section, we will show the design and functional phase of our application. The user accesses the web application where the platform is hosted and registers as well as votes securely and transparently. The methodology is given below.

3.3.1 Registration Stage

In this stage, the voter has to register with his unique Aadhar number, email address, name, and phone number. They also need to add their clear photo here as this system uses facial recognition technology to log in.

3.3.2 Login

After registration, the voter tries to log in and cast his vote. At this stage, the voter first logs in using a password. After successful login, the voter must authenticate to vote. For real-time authentication, facial recognition technology is used to increase the level of security.

3.3.3 Blockchain Technology

This technology is mainly used for its security features. Blockchain provides a secure and transparent environment. Blockchain encrypts the voter's message (Casted vote) using an asymmetric encryption algorithm. The public key is provided by the Blockchain and the private key is with the host. The public key is used for authentication purposes using the ledger.

3.3.4 Ethereum

The Ethereum network provides a framework for creating and storing the blockchain. Each block is created and its details are stored in an encrypted ledger. These generated blocks are distributed among the nodes, providing the system with high fault tolerance. To cast a vote, a user must use Ethereum for their transactions. So, Ganache is used for this. Ganache is a private Ethereum blockchain environment that allows you to emulate the Ethereum blockchain to interact with smart contracts on your private blockchain.

3.3.5 Database

All system data is stored in the MongoDB database. The data will be in the form of voter and candidate names, unique voter IDs, and voting details like time, time slot, region, etc.

3.3.6 Admin

Admin will control the entire environment. Verification of voters and candidates will be done by admin. Admin only arranges the voting schedule. All important notifications like results etc are also under admin control.

3.3.7 Results Phase

The processing and counting of votes take place in the results phase. The results are generated and displayed on the website. Users can verify their votes using their public key. This ensures the transparency of the voting system.

3.3.8 Meta Mask

Metamask allows blockchain users to manage their wallets. Using the browser extension, users can use the wallet and perform transactions through the browser. When a transaction is performed, a meta mask pops up and asks the user to confirm the transaction.

3.3.9 Truffle

Truffle offers an improved environment primarily based totally on the Ethereum blockchain. Truffle is capable of compiling the Ethereum contracts and migrating them. After migration contracts are deployed on ganache, any Ethereum takes a look at the net (e.g., Ropsten, Rinkeby, local network) or on an actual Ethereum network.

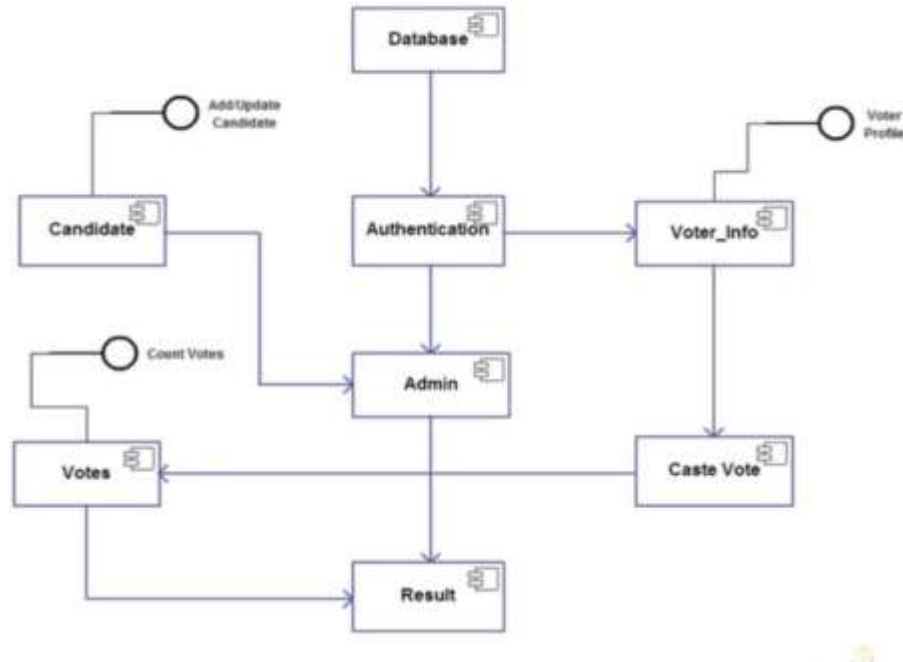


Figure 9 Overview of System

3.4 Voting Process

We now describe a typical user interaction with the proposed scheme based on our current system implementation. So basically, the voter logs into the system by scanning their face. After scanning the face, the facial recognition system authenticates the voter. If a match is found, the voter is presented with a list of available candidates with the option to vote against them. Conversely, if the match is unsuccessful, any further access would be denied.

This functionality is achieved by using an appropriate implementation of an authentication mechanism (in this case a facial recognition system) and predefined role-based access control. Furthermore, it is also assumed that the voter is assigned to his particular electoral district and this information is used to create a list of candidates for which the voter can vote. Assigning a voter to a constituency is considered an offline process and is therefore beyond the scope of this research.

After successfully casting votes, it is mined by multiple miners for verification, after which valid and verified votes are added to the public ledger. The security aspects of voting are based on blockchain technology using cryptographic hashes to secure end-to-end verification.

For this purpose, a successfully cast vote is considered a transaction within the voting application's blockchain. Therefore, the casting vote is added as a new block (after successful mining) in the blockchain and is also recorded in the data tables at the end of the database. The system ensures ownership of voting systems for only one person and one vote.

This is achieved by using a unique voter face that matches at the start of each voting attempt to prevent double voting. Once miners mine a vote, a transaction is generated that is unique to each vote. If the vote is found to be malicious, the miner is rejected.

After the validation process, a notification is immediately sent to the voter via message or email with the transaction ID defined above, through which the user can track their vote to the ledger. While this works as a voter notification, it does not allow any user to extract information about how a particular voter voted, thereby achieving voter privacy.

It is important to note here that the cryptographic hash for the voter is the unique hash of the voter by which the voter is known in the blockchain. This feature makes it easier to achieve verifiability of the entire voting process. In addition, this id is hidden and no one can view it, not even the system operator or administrator can view this hash, thereby achieving the privacy of individual voters.

3.5 Blockchain

3.5.1 Introduction

Blockchain technology is shining sort of a star these days when its entry and wide-spread adoption of Bitcoin, the terribly initial cryptocurrency that involves people's minds. Blockchain technology originates from the fundamental subject style of the bitcoin cryptocurrency, wherever it absolutely was initial introduced to the net world and previously became a promising technology thanks to the high degree of transparency within the system, turning into a vigorous space of analysis and study for its varied applications. alternative fields.

Blockchain, simply put, may be a shared, changeless ledger that facilitates the method of recording transactions and following assets in a very business network.

Associate in Nursing quality are often tangible (house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). during this era nearly we are able to track and trade something on a blockchain network, reducing prices and risk for everybody.

Blockchain stores its knowledge in blocks. First, all the info to be kept within the blockchain is reborn into smaller components, that square measure allotted to totally different blocks within the suburbanized network. The initial block in a very blockchain is understood as a "Genesis Block" or "Block 0". "Block Gene-sis" or "Block 0".

The genesis block is sometimes hard-coded into software; is peculiar therein it doesn't contain a link to the previous block. The genesis block is sometimes hard-coded into the software; it's special therein it doesn't contain a regard to the previous block (the Genesis block).

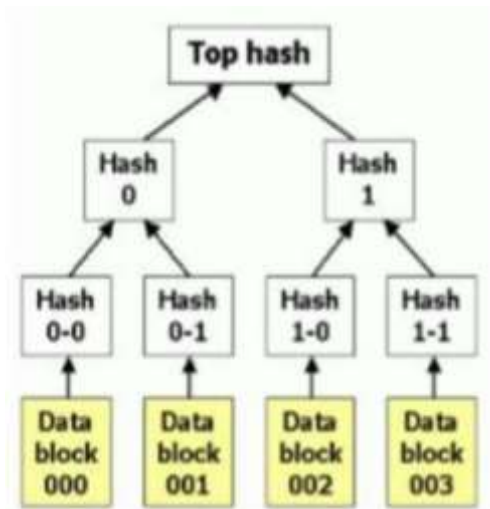


Figure 10 Hash Table

Once the genesis block is initialized, "Block 1" is created and when completed is attached to the genesis block. Each block has a transaction data part, a copy of each of the transactions is hashed and then the hashes are matched and hash again, this continues until there is only one hash left; to known as the Merkle root (Figure 10).

The block header is where the Merkle root is stored, which ensures that the transaction cannot be modified by third parties.

Blockchain is designed to be accessed through a peer-to-peer network, each node then communicates with other nodes to exchange blocks and transactions. A blockchain block consists of a block header, a hash value of the previous block header, and a Merkle root.

So, when we extract data from the blockchain, then all the smaller parts that are in the decentralized network are connected by accessing all the blocks through their hash values. If a person or a third party wants to change the data, they need to know the hash values of all the tables, without them they can't take even a bit of information about our data. Because these data blocks are distributed in thousands and thousands of blocks, it takes a hacker more than millions of years to find the hash values of all the blocks. Thanks to these features, it provides safety, reliability, and robustness.

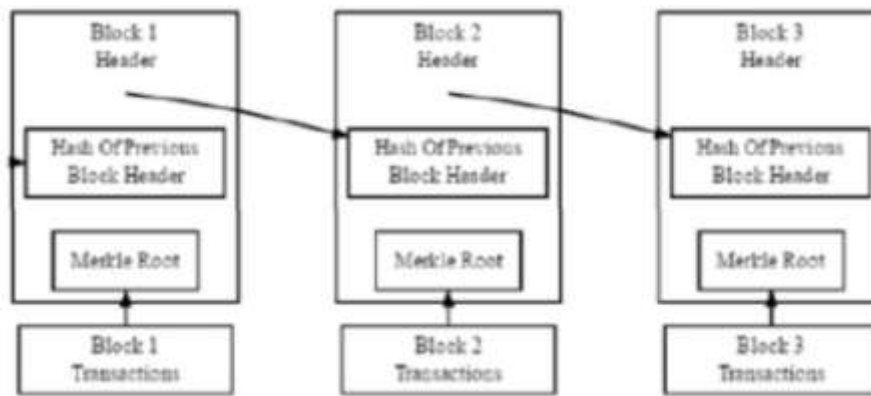


Figure 11 Simplified Bitcoin Blockchain

Before 2004 there was a paper-primarily based totally vote casting device referred to as ballot paper device in India. So, to eliminate all the shortcomings of the traditional voting system, the blockchain fits as the best suitable solution for the e-voting platform. Electronic voting is extensively studied and many implementations are tested and even used for a while where we encountered some problems like user authentication, vote tempering, etc.

Of course, there are many government websites like polls, networks to gain knowledge about new government schemes, questionnaires where common people raise their questions, etc. still we can't say the same for online elections because every vote matters here for a candidate so the system needs to be more secure, reliable and authentic.

This is mainly because official elections are the basic elements of democracy and democratic governance which is the most preferred administrative methodology in the modern world.

The electronic voting system will be a new change from the regular voting system, which is less

complicated, and more open to the voters, and besides, security will come first. These tactics reduce the cost of many laws to some voters by creating more ways for them to cast a ballot.

This gives it a chance to survive long queues at check stations and offers a better experience to those who are sick, serving in the military, people living abroad or gone abroad on holiday, etc. It has been found that many people did not vote due to laziness, so it can serve its purpose here in the electronic voting system. And all the things in today's world are digitized, then the electoral system falls behind, young people between the ages of 18 and 30 are special voters, and the web is a process to attract those citizens who seem to be the hardest to reach.

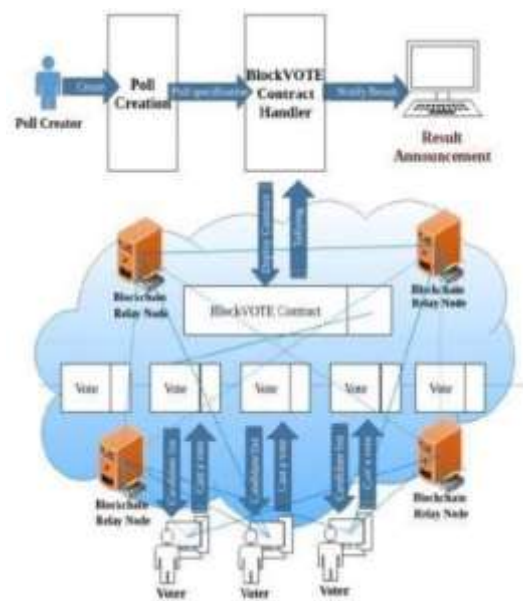


Figure 12 Architecture of E-Voting

Our main motivation behind this project is to create a secure voting environment and show that a reliable e- voting scheme is possible using blockchain. Because when electronic voting is available to anyone with a computer or mobile phone, every single administrative decision can be made by people and members; or at least people's opinions will be more public and accessible to politicians and managers. This will eventually lead humanity to true direct democracy.

This is important to us because elections can be easily corrupted or rigged, especially in small towns and even larger cities in corrupt countries. Moreover, large-scale traditional elections are very expensive in the long run, especially when there are hundreds of geographically distributed polling stations and millions of voters.

Voter turnout at polling stations is also relatively low, as the person may not be staying at the address whose name is on the list, or maybe on vacation or other work. Electronic voting will be able to solve these problems if implemented carefully. The concept of electronic voting is significantly older than blockchain.

So, all known examples to date have used the means of centralized computing and storage models. By implementing blockchain, we can increase the security level of our system due to its way of storing data in a decentralized network.

This project also has a facial recognition system to verify whether the user is valid or not. Some systems were found to have an OTP system to authenticate the user. But in this authentication model, the disadvantage is that if x person's phone is not with him, even though his phone is with his friend, his friend can vote twice because of the OTP system, one with the original chance and one because of his friend's login, and then x can't vote. So here comes the face recognition technique where the user has to verify his face before voting to maintain transparency.

Estonia is a very good example as the Estonian government is one of the first to implement a fully online and comprehensive e-voting solution. The concept of electronic voting began to be discussed in the country in 2001 and was officially launched by the national authorities in the summer of 2003. Their system is still in use, with many improvements and modifications to the original scheme.

As mentioned, it is currently very robust and reliable. They use smart digital ID cards and personal card readers (distributed by the government) for personal authentication. There is a dedicated web portal and an equivalent desktop application for citizens to participate in the elections by nominating candidates and casting their votes. So that anyone with a computer, an internet connection, and also an ID card can easily vote remotely.

Switzerland is another of the few countries participating in the electronic voting trend. In Switzerland, known for its widespread democracy, every citizen who reaches the age of 18 can actively or passively participate in elections, which can be held on many different topics for many different decisions. They also started official work on a voting system called remote voting.

Case in point – March 2018 general elections in Sierra Leone – Swiss startup Agora conducted a count in two districts. After the vote, a team of accredited observers from various locations manually entered approximately 400,000 ballots into Agora's blockchain system. This system was a partial deployment of the blockchain and the voting was verified by the blockchain and not powered by the blockchain.

3.5.2 Solidity

Solidity is an object-oriented, high-level programming language used to create smart contracts that automate transactions on the blockchain. After being proposed in 2014, the language was developed by contributors to the Ethereum project.

The language is primarily used to create smart contracts on the Ethereum blockchain and create smart contracts on other blockchains.

Solidity is similar to one of the most common programming languages, JavaScript. It can be considered as a dialect of JavaScript. This means that if you understand JavaScript, it can be easy to pick up Solidity. Solidity also shares similar characteristics to the programming languages C++ and Python.

As a high-level language, Solidity does away with the need to type code in ones and zeros. It makes it much easier for humans to write programs in ways they find easier to understand, using a combination of letters and numbers.

Solidity is statically typed, with support for inheritance, libraries, and complex user-defined types. As Solidity is statically typed, the user must specify each variable. Data types allow the compiler to check for the correct use of variables. Solidity data types are usually categorized as either value types or reference types.

The main difference between value types and reference types can be found in how they are assigned to a variable and stored in the EVM (Ethereum Virtual Machine). While changing the value in one variable of a value type does not affect the value in another variable, anybody referring to changed values in reference type variables may get updated values.

• How Does Solidity Work?

The beauty of the Ethereum ecosystem is that so many different cryptocurrencies and decentralized applications can use it. Smart contracts make it possible for unique technologies

to be made on Ethereum for all kinds of businesses and organizations.

Every year, the world spends billions of dollars on blockchain solutions. Many of these solutions are created using Solidity. Smart contracts built using Solidity can be thought of as a way to automate business and non-business processes between different people. This ensures that people making transactions on the blockchain.

3.5.3 Smart Contract

A smart contract, like any contract, establishes the terms of an agreement. But unlike a traditional contract, a smart contract's terms are executed as code running on a blockchain like Ethereum. Smart contracts allow developers to build apps that take advantage of blockchain security, reliability, and accessibility while offering sophisticated peer-to-peer functionality everything from loans and insurance to logistics and gaming.

3.5.3.1 How do smart contracts work?



Figure 13 Smart Contract Working

Smart contracts are written in a variety of programming languages (including Solidity, Web Assembly, and Michelson). On the Ethereum network, each smart contract's code is stored on the blockchain, allowing any interested party to inspect the contract's code and current state to verify its functionality.

Each computer on the network (or “node”) stores a copy of all existing smart contracts and their current state alongside the blockchain and transaction data.

When a smart contract receives funds from a user, its code is executed by all nodes in the network in order to reach a consensus about the outcome and resulting flow of value. This is what allows smart contracts to securely run without any central authority, even when users are making complex financial transactions with unknown entities.

To execute a smart contract on the Ethereum network, you will generally have to pay a fee called “gas” (so named because these fees keep the blockchain running).

3.5.4 Ethereum

Ethereum is a technology for building apps and organizations, holding assets, transacting and communicating without being controlled by a central authority. There is no need to hand over all your personal details to use Ethereum - you keep control of your own data and what is being shared. Ethereum has its own cryptocurrency, Ether, which is used to pay for certain activities on the Ethereum network. Here are some benefits of Ethereum.

3.5.4.1 Large, existing network. The benefits of Ethereum are a tried-and-true network that has been tested through years of operation and billions of value trading hands. It has a large and committed global community and the largest ecosystem in blockchain and cryptocurrency.

3.5.4.2 Wide range of functions. Besides being used as a digital currency, Ethereum can also process other financial transactions, execute smart contracts and store data for third-party applications.

3.5.4.3 Constant innovation. A large community of Ethereum developers is constantly looking for new ways to improve the network and develop new applications. “Because of Ethereum’s popularity, it tends to be the preferred blockchain network for new and exciting (and sometimes risky) decentralized applications,” Avital says.

3.5.4.4 Avoids intermediaries. Ethereum’s decentralized network promises to let users leave behind third-party intermediaries, like lawyers who write and interpret contracts, banks that are intermediaries in financial transactions or third-party web hosting services.

Ethereum Disadvantages:

3.5.4.5 Rising transaction costs. Ethereum's growing popularity has led to higher transaction costs. Ethereum transaction fees, also known as "gas," can fluctuate and be quite costly. That's great if you're earning money as a miner but less so if you're trying to use the network. Unlike Bitcoin, where the network rewards transaction verifiers, Ethereum requires those participating in the transaction to cover the fee.

3.5.4.6 Potential for crypto inflation. While Ethereum has an annual limit of releasing 18 million Ether per year, there's no lifetime limit on the potential number of coins. This could mean that as an investment, Ethereum might function more like dollars and may not appreciate as much as Bitcoin, which has a strict lifetime limit on the number of coins.

3.5.4.7 Steep learning curve for developers. Ethereum can be difficult for developers to pick up as they migrate from centralized processing to decentralized networks.

3.5.5 Metamask

MetaMask is a cryptocurrency wallet that is available as a browser extension to help you store tokens, interact with decentralized applications, and trade Ethereum. By connecting users with MyEtherWallet, MetaMask eliminates the need to enter private keys at the execution of each transaction while creating, storing, or trading tokens.

Users can store and manage their Bitcoin, Ether, and other cryptocurrencies using a blockchain wallet, which is available as a digital or online wallet. A blockchain wallet enables cryptocurrency transfers, prevents theft of crypto assets, and allows users to convert them back into their local currencies if needed. Its community is strong, with over a million downloads, and it has many resources contributing to the project.

The MetaMask browser extension is a browser extension that can act as your Ethereum wallet. Unlike traditional wallets, MetaMask requires no additional plug-ins, so you're free to use it in any browser.

Once installed, you can view your Ethereum address and send or receive coins to any Ethereum address in turn. MetaMask allows you to stake coins on gambling sites and trade on DEXs, as well as participate in projects like Pool Together and Compound.

3.5.6 Truffle

Truffle is the development environment, asset pipeline, and testing framework of the Truffle Suite ecosystem. Truffle is a hugely popular development framework for Ethereum dApp

development, and there is a large community behind the tool. Furthermore, Truffle is using the EVM as a basis, and one of its purposes is to make the development of smart contracts more straightforward and more accessible.

3.5.7 Ganache

Ganache is a tool that allows us to spin up our own local Ethereum blockchain. The blockchain can be utilized throughout all parts of the development process, making this tool highly useful. As we set up our local blockchain, Ganache allows us to deploy, develop and test all our dApps in a safe and deterministic environment.

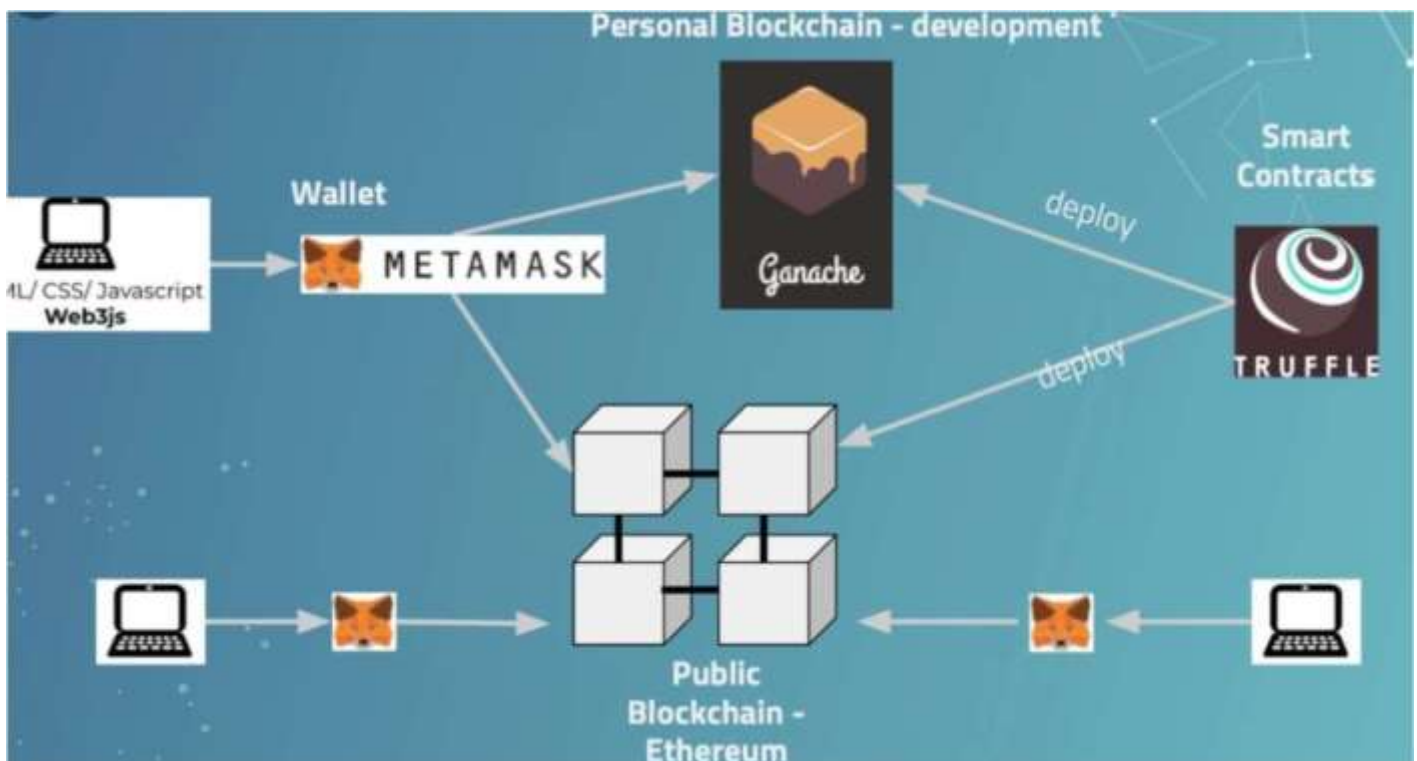


Figure 14 Blockchain Transaction Flow

3.6 Face Recognition

User Identification is done here with the help of face recognition technique. In order to cast vote to any candidate, user need to first authenticate by using face recognition. Face Recognition will ensure reliability that the person who is voting is a validate one and no other person is voting in spite of him/her.

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labelled Faces in the Wild benchmark. This also provides a simple face recognition command line tool that lets you do face recognition on a folder of images from the command line.

3.6.1 Features

3.6.1.1 Find face



Figure 15 Find Face

3.6.1.2 Find and manipulate facial features

Get the locations and outlines of each person's eyes, nose, mouth and chin.

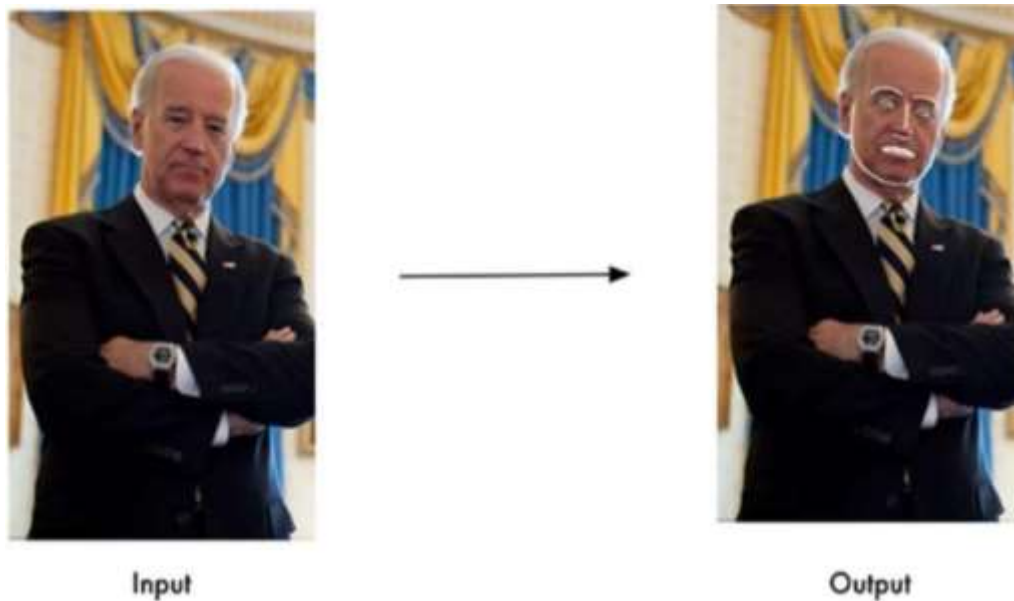


Figure 16 Extract Facial Feature

3.6.1.3 Identify faces in pictures

Recognize who appears in each photo.

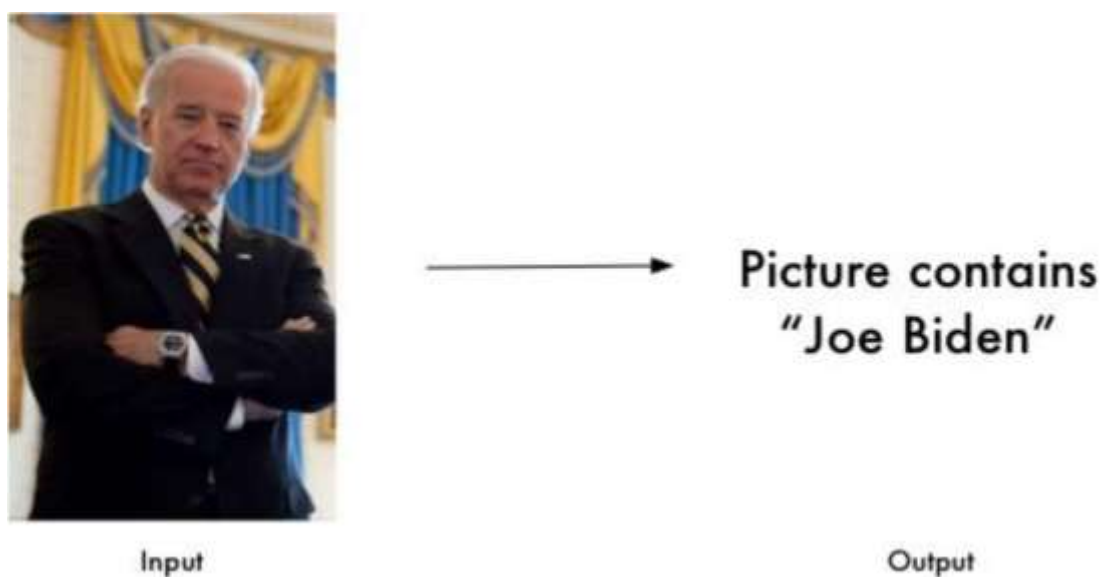


Figure 17 Identify Faces

3.6.2 Command-Line Interface

When you install face_recognition, you get a simple command-line program Called face_recognition that you can use to recognize faces in a photograph or folder full for photographs. First, you need to provide a folder with one picture of each person you already know.

There should be one image file for each person with the files named according to who is in the picture. Next, you need a second folder with the files you want to identify

Then in you simply run the command face_recognition, passing in the folder of known people and the folder (or single image) with unknown people and it tells you who is in each image.

There's one line in the output for each face. The data is comma-separated with the filename and the name of the person found. An unknown_person is a face in the image that didn't match anyone in your folder of known people.

3.7 Timeline Chart

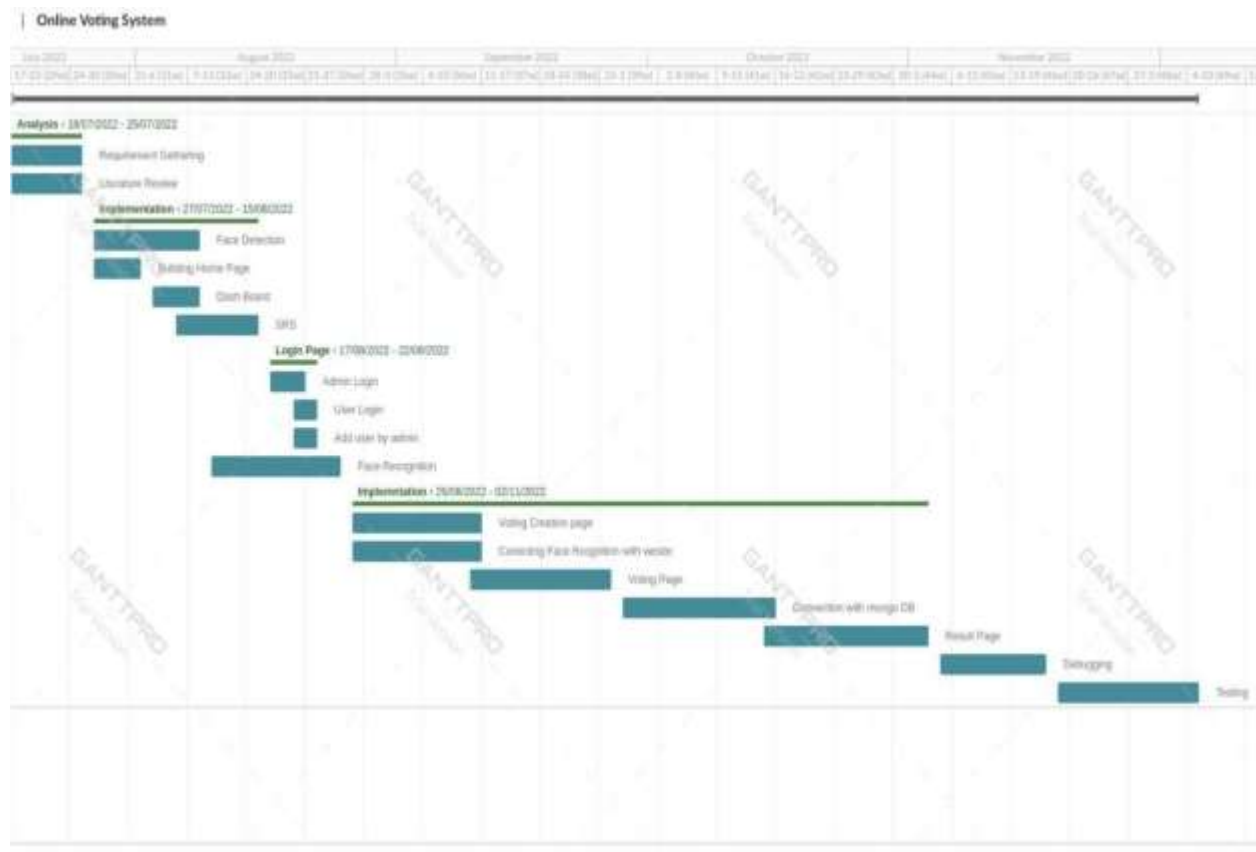


Figure 18 Project Timeline Chart

CHAPTER 4

4. IMPLEMENTATION AND TESTING

4.1 Software and Tools

4.1.1 Hardware Requirements

4.1.1.1 Hardware that supports web app (E.g., Mobile, Computer)

Hardware must be there until it is hosted to cloud servers and ready to pitch. After that anyone can use it through any smartphones or pc that has web browser inside.

4.1.2 Software Requirements

4.1.2.1 Front-End: React-js

4.1.2.2 React is a JavaScript library for building user interfaces.

4.1.2.3 React is used to build single-page applications.

4.1.2.4 React allows us to create reusable UI components.

4.1.2.5 Back-End: Solidity

4.1.2.6 Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behavior of accounts within the Ethereum state.

4.1.2.7 Solidity is a Curly bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript. You can find more details about which languages Solidity has been inspired by in the language influences section.

4.1.2.8 Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

4.1.2.9 With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi- signature wallets.

4.1.2.10 When deploying contracts, you should use the latest released version of Solidity. Apart from exceptional cases, only the latest version receives security fixes.

Furthermore, breaking changes as well as new features are introduced regularly. We currently use a 0.y.z version number to indicate this fast pace of change

4.1.2.11 Others: Domain name, Hosting, sever and cloud storage

This web application needs to be computer under solidity language as it is the Blockchain technology supported as well the latest front-end language React-js that should be pre-installed in hardware in complete working position.

4.1.3 Software requirements for our clients

4.1.3.1 Windows 7 or higher OS

4.1.3.2 Google chrome or any other safe browser

Client just need not to worry about anything just trust on over new innovation and head towards the transaction through any smartphone or pc that has web browser inside.

4.2 User Interface and Snapshots

4.2.1 Admin Login

Admin Login used for admin to Login in the System without email address and password admin or any other person cannot login using this module as shown in Figure 19.

Figure 19 Admin Login

4.2.2 Admin Dashboard

Here Figure 20 shows Admin Dashboard where admin get all the information like total number of users, candidates and elections available in the system and also navigate to other modules like user candidate, election changing phase or showing result.

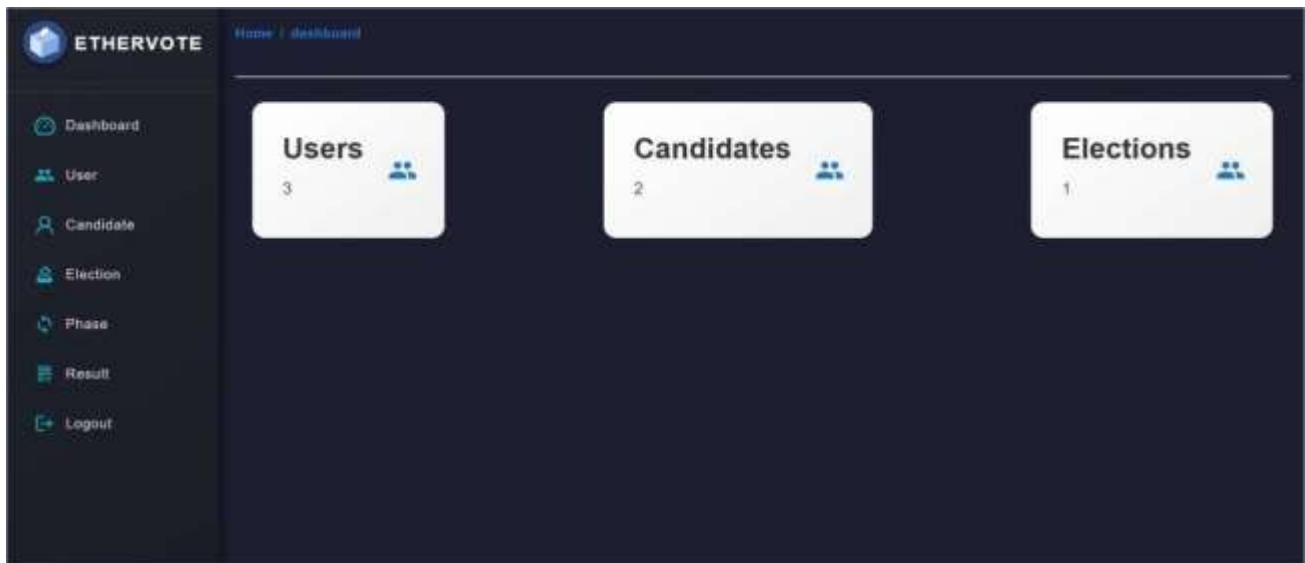


Figure 20 Admin Dashboard

4.2.3 View Users

Admin can view details of the users as shown in Figure 21. he can perform operations like edit users or delete user and add the user in the voting system.

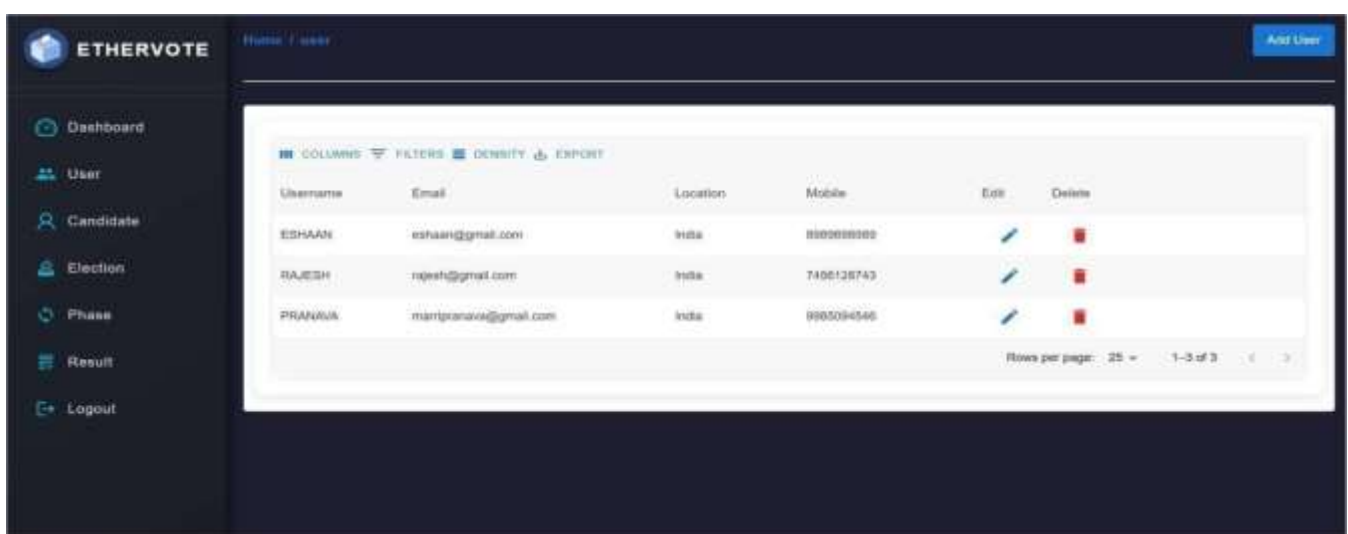


Figure 21 View Users

4.2.4 Add User

Admin can add user in the system using this gui where admin have to fill details like username, first name and other details after click on add user guy will get username and password in the e-mail which is entered in the given form.

The screenshot shows a web application interface with a dark sidebar on the left containing navigation links: Dashboard, User, Candidate, Election, Phase, Result, and Logout. The main content area displays a light-colored form titled 'ADD USER'. The form includes the following fields: a single-line text input for 'Username'; two single-line text inputs for 'First Name' and 'Last Name'; a single-line text input for 'E-mail'; a single-line text input for 'Mobile'; two single-line text inputs for 'Password' and 'Confirm Password'; and a file upload section with a 'Choose file' button and the text 'No file chosen'. A prominent blue button labeled 'ADD USER' is positioned at the bottom center of the form.

Figure 22 Add User

4.2.5 Add Candidate

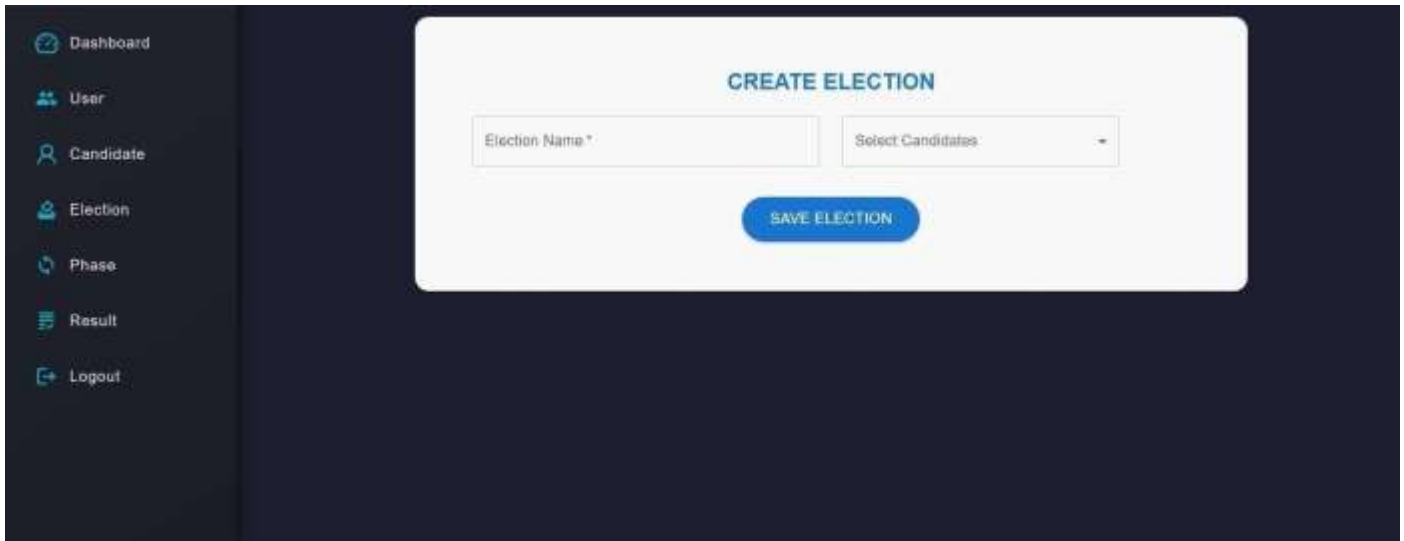
Admin can add candidate and in the voting phase user will vote for this candidate.

The screenshot shows the same web application interface as Figure 22. The sidebar navigation is identical. The main content area displays a light-colored form titled 'ADD CANDIDATE'. The form includes the following fields: a single-line text input for 'Username'; two single-line text inputs for 'First Name' and 'Last Name'; a 'Birth Date' field with a date picker icon; a 'Pictore' field with a file upload icon; another 'Pictore' field with a file upload icon; a single-line text input for 'Qualification'; a single-line text input for 'Location'; and a multi-line text area for 'Description'. A prominent blue button labeled 'ADD CANDIDATE' is positioned at the bottom center of the form.

Figure 23 Add Candidate

4.2.6 Add Election

Admin can add new election in the system with the unique election name and also add candidate from the available candidate with the dropdown menu of candidates.

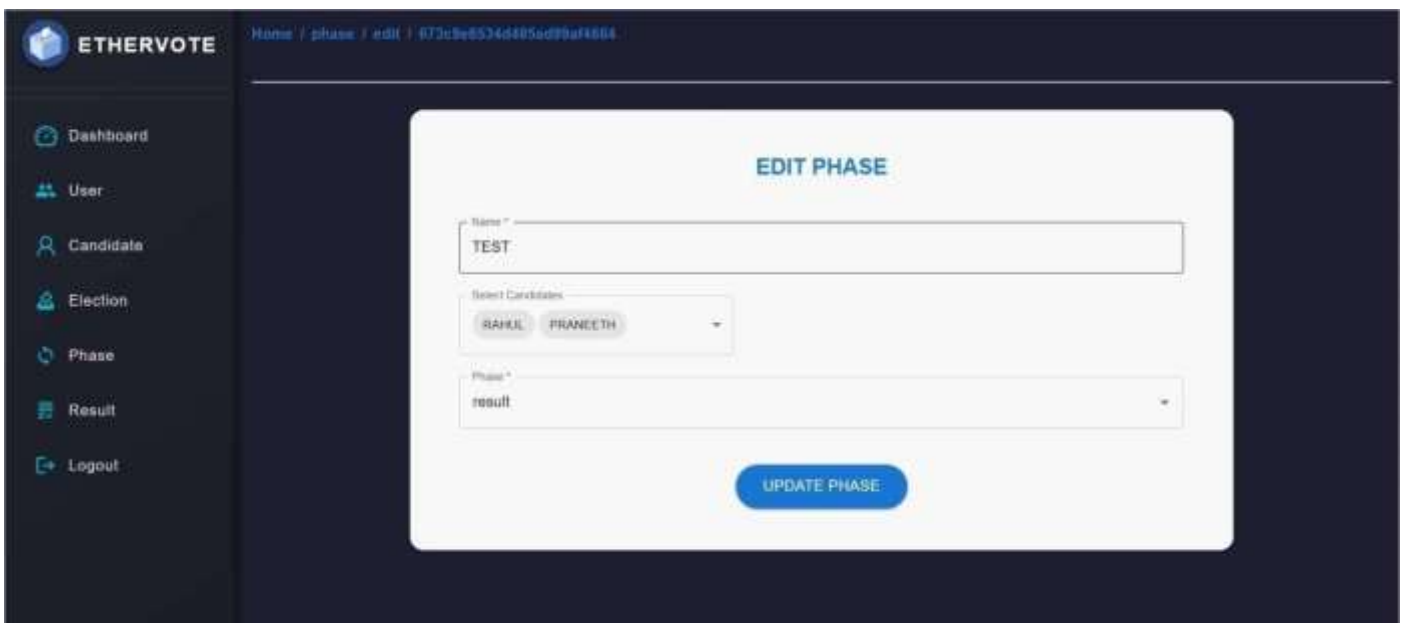


The screenshot shows the 'CREATE ELECTION' form within a dark-themed dashboard. On the left is a sidebar with navigation links: Dashboard, User, Candidate, Election, Phase, Result, and Logout. The main content area features a white card with the title 'CREATE ELECTION'. Inside the card, there is a text input field labeled 'Election Name *' and a dropdown menu labeled 'Select Candidates:'. Below these fields is a blue button labeled 'SAVE ELECTION'.

Figure 24 Add Election

4.2.7 Edit Phase

From this Page Admin can change the phase of election.



The screenshot shows the 'EDIT PHASE' form within the 'ETHERVOTE' dashboard. The sidebar on the left includes the 'ETHERVOTE' logo and navigation links: Dashboard, User, Candidate, Election, Phase, Result, and Logout. The main content area displays a white card titled 'EDIT PHASE'. The form contains three fields: a text input for 'Name *' with the value 'TEST', a dropdown for 'Select Candidates' showing 'RAHUL' and 'PRANEETH', and a dropdown for 'Phase *' with the value 'result'. A blue 'UPDATE PHASE' button is positioned at the bottom right of the card. The top of the page shows a breadcrumb trail: 'Home / phase / edit / 872c9e6534d485ad99af4664'.

Figure 25 Edit Phase

4.2.8 View Elections

Here User can select election in which election user want to vote.



Figure 26 Election

4.2.9 Candidate of Election

Here user can select candidate for voting and click on vote the python script will be execute and camera will be open in 30 seconds.

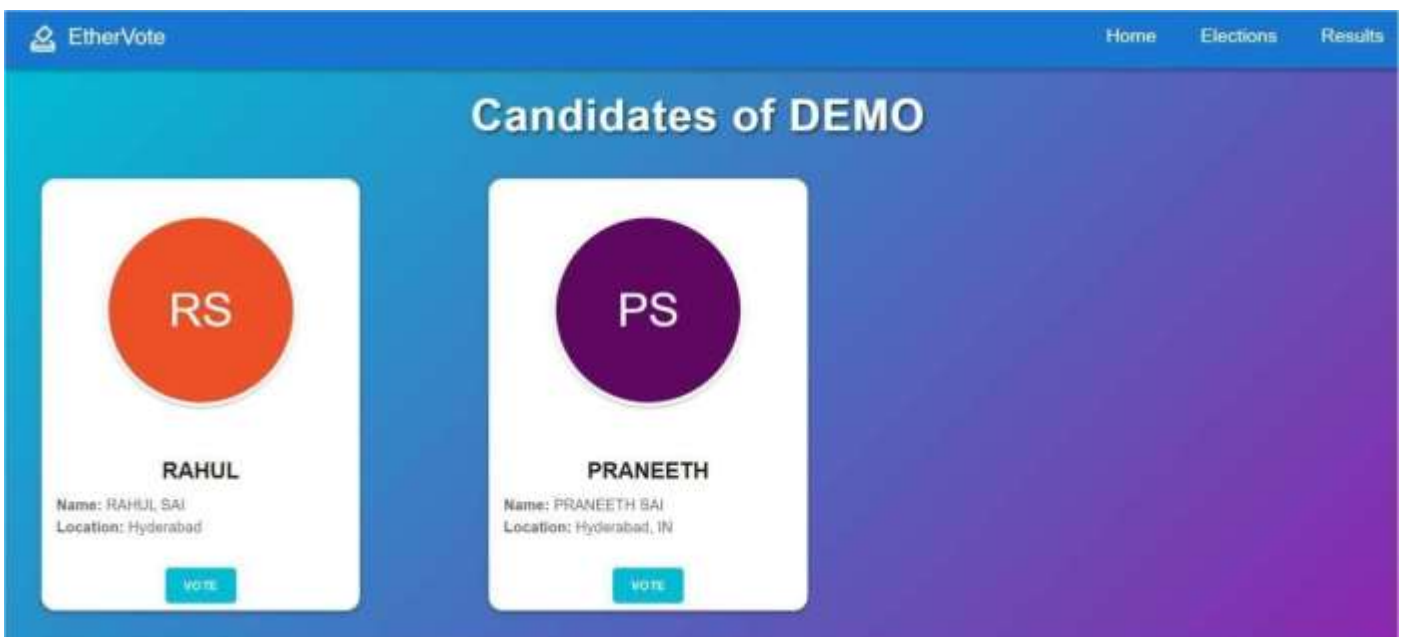


Figure 27 Candidates of Elections

4.2.10 Login User

User will be recognize using python script and the username will be shown in the username box automatically After entering right password metamask will be open for transaction.



The screenshot displays the 'EtherVote' user login interface. At the top, a blue header bar contains the 'EtherVote' logo on the left and navigation links for 'Home', 'Elections', and 'Results' on the right. Below the header, the page title 'Enter Credentials' is centered. The login form consists of four input fields: 'Username *' with the value 'PTANAVA', 'Enter id *' with a masked value 'xxxxxxxxxxxxxxxx', 'Candidate Name *' with the value 'RAHUL', and 'Password *'. A blue 'VOTE' button is positioned at the bottom left of the form area.

Figure 28 User Login Page

4.2.11 Metamask Transaction Details

Here Ethereum will be transfer from user account to candidate account using metamask After confirming transaction user will get alert for successful transaction and also get email.

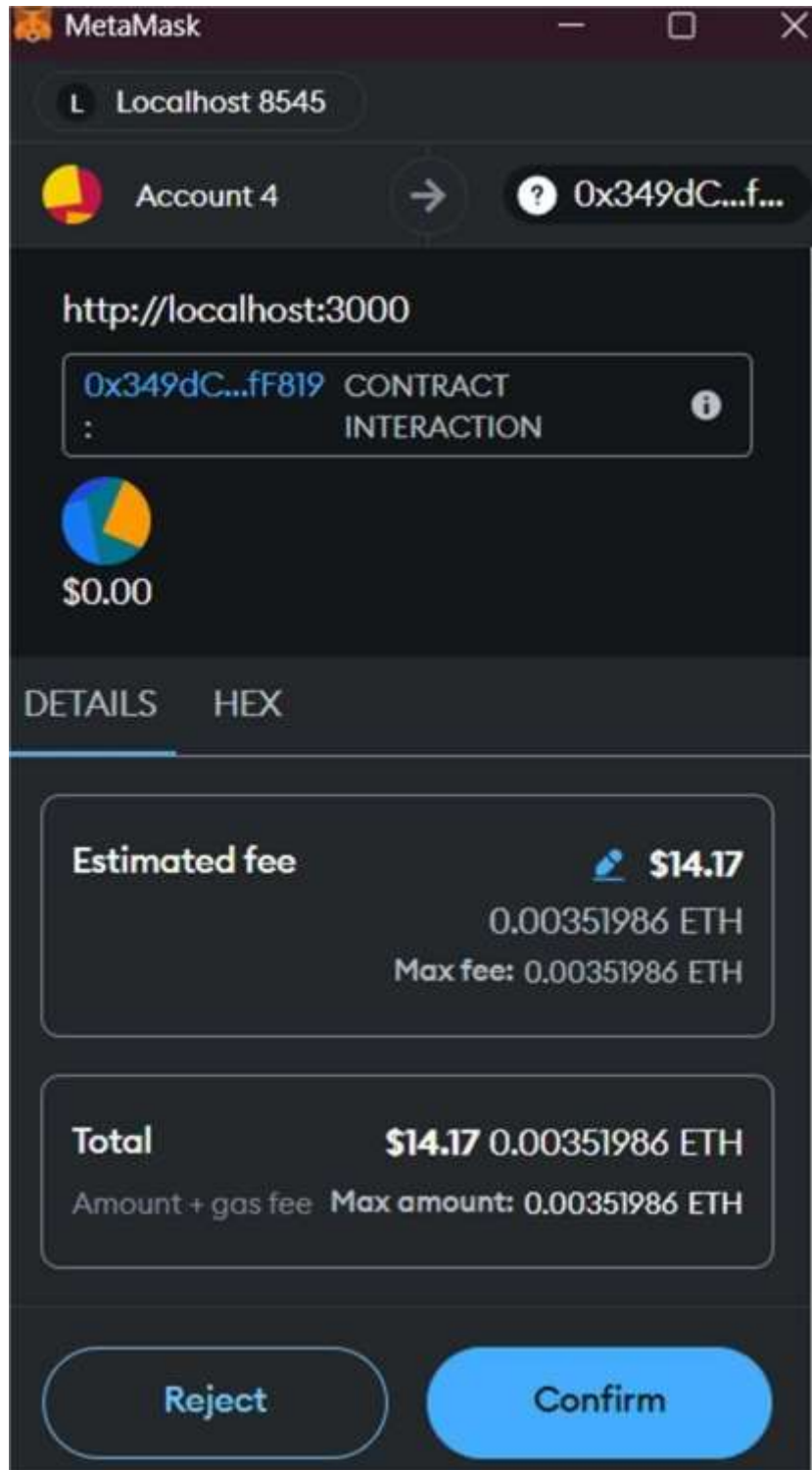


Figure 29 Transaction Details

4.2.12 Result of Elections

Here user can view all the election which phase is result and view details of the elections.

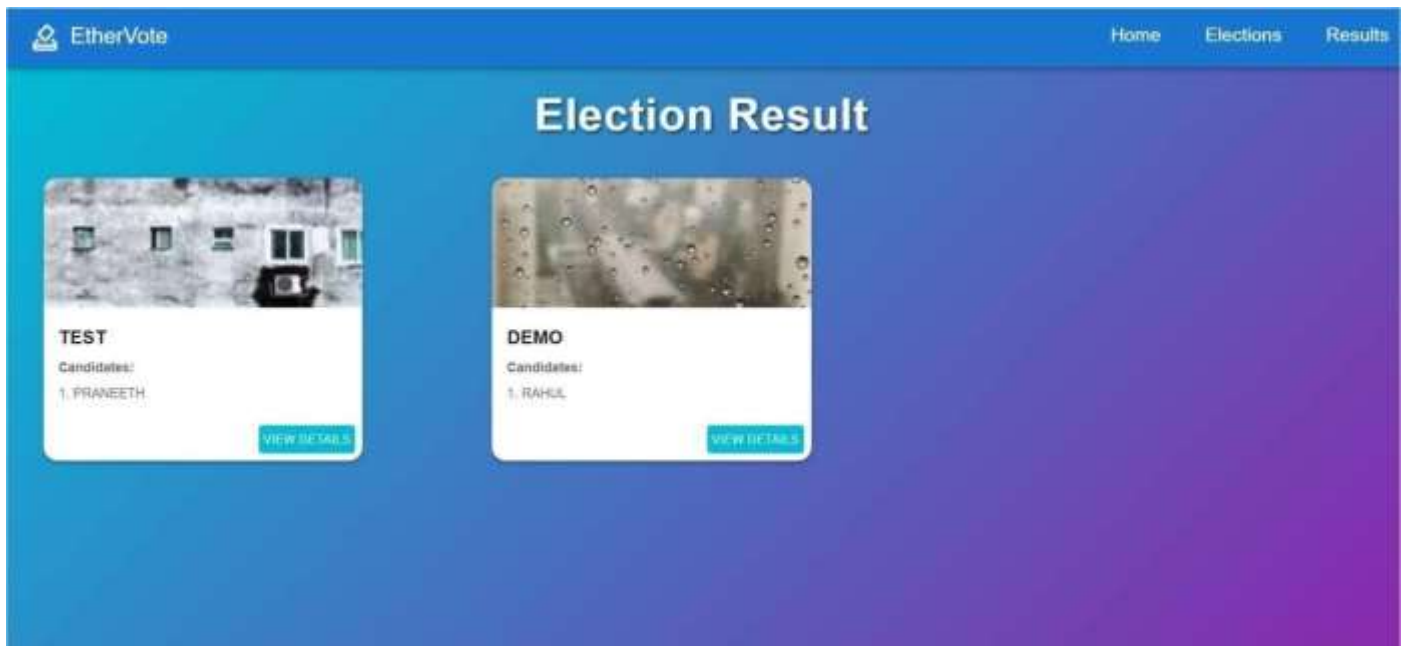


Figure 30 Result of Elections

4.2.13 Result of Candidates

Here user can view total vote of the particular candidate which election is chosen.



Figure 31 Candidate Result

CHAPTER 5

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

To conclude in nutshell, by using this project, the user doesn't need to worry about any special tool to write in the open space. With optimum use of resources, we can write and convey correct message to other party.

In the current situation, online work is unavoidable. Mainly, online education is proving harder as teachers sometimes feel difficult to teach without writing. This is where our project comes into picture.

Once the project website is hosted on server, anyone (i.e., teachers) can access it and using the web-camera one can easily do their task using writing. (i.e., Math teacher may use it to show equations to the students in online education)

To sum it up, we can say that using python libraries and OpenCV, we have created a tool through which all people who were unable to convey the message properly due to lack writing in real time during online meetings can now do it efficiently.

5.2 Future work

As this system can be mainly used for college voting so its part of face recognition can be used as an attendance system where we just have to add the scanned student name into a file.

This system uses Ethereum coins for casting the votes so in future we can use Hyperledger (i.e., it doesn't use a cryptocurrency for transaction) for implementing blockchain so we can make this system free for people to cast votes.

CHAPTER 6

6 REFERENCES

1. R. Taş and Ö. Ö. Tanrıöver, "A systematic review of challenges and opportunities of blockchain for E- voting", *Symmetry*, vol. 12, no. 8, pp. 1328, Aug. 2020.
2. 5. Onuklu, A. (2019), "Research on Blockchain: A Descriptive Survey of the Literature", Choi, J. and Ozkan, B. (Ed.) *Disruptive Innovation in Business and Finance in the Digital World* (International Finance Review, Vol. 20), Emerald Publishing Limited, pp. 131-148. DOI/10.1108/S1569-3767201
3. Zhang K, Zhang Z, Li Z, et al. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks [J]. *IEEE Signal Processing Letters*, 2016, 23(10):1499-1503.
4. Pranav KB, Manikandan J, " Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks", April 2020, ScienceDirect
5. Shahzad, B.; Crowcroft, J. Trustworthy Electronic Voting Using Adjusted Blockchain Technology. *IEEE Access* 2019, 7, 24477–24488.
6. Gao, S.; Zheng, D.; Guo, R.; Jing, C.; Hu, C. An Anti-Quantum E-Voting Protocol in Blockchain with Audit Function. *IEEE Access* 2019, 7, 115304–115316.
7. Ramya Govindaraj, P Kumaresan, K. Sree harshitha, " Online Voting System using Cloud," 24-25 Feb. 2020, IEEE
8. Fernández-Caramés, T.M.; Fraga-Lamas, P. Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks. *IEEE Access* 2020, 8, 21091–21116.
9. Yi, H. Securing e-voting based on blockchain in P2P network. *EURASIP J. Wirel. Commun. Netw.* 2019, 2019, 137.
10. Torra, V. Random dictatorship for privacy-preserving social choice. *Int. J. Inf. Secur.* 2019, 19, 537– 543.
11. Alaya, B.; Laouamer, L.; Msilini, N. Homomorphic encryption systems statement: Trends and challenges. *Comput. Sci. Rev.* 2020, 36, 100235.