



MINI PROJECT REPORT

On

TRUEVOTE

Submitted in partial fulfilment for the award of degree

Of

Master of Computer Applications

By

AMAL SHAJI (MLM24MCA-2008)

Under the Guidance of

Ms. ANJU JOHN

(Assistant Professor, Department of Computer Applications)



DEPARTMENT OF COMPUTER APPLICATIONS

MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR

(Affiliated to APJ Abdul Kalam Technological University)

OCTOBER 2025



MANGALAM COLLEGE OF ENGINEERING
Accredited by NAAC & ISO 9001:2000 Certified Institution
DEPARTMENT OF COMPUTER APPLICATIONS

VISION

To become a centre of excellence in computer applications, competent in the global ecosystem with technical knowledge, innovation with a sense of social commitment.

MISSION

- To serve with state of the art education, foster advanced research and cultivate innovation in the field of computer applications.
- To prepare learners with knowledge skills and critical thinking to excel in the technological landscape and contribute positively to society.

Program Educational Objectives

- **PEO I** : Graduates will possess a solid foundation and in-depth understanding of computer applications and will be equipped to analyze real-world problems, design and create innovative solutions, and effectively manage and maintain these solutions in their professional careers.
- **PEO II**: Graduates will acquire technological advancements through continued education, lifelong learning and research, thereby making meaningful contributions to the field of computing.
- **PEO III**: Graduates will cultivate team spirit, leadership, communication skills, ethics, and social values, enabling them to apply their understanding of the societal impacts of computer applications effectively.

Program Specific Outcomes

- **PSO I**: Apply advanced technologies through innovations to enhance the efficiency of design development.
- **PSO II**: Apply the principles of computing to analyze, design and implement sustainable solutions for real world challenges.

MAPPING OF PO-PSO-SDG

1. MAPPING WITH PROGRAM OUTCOMES (POs):-

SL.NO	POs ADDRESSED	RELEVANCE TO PROJECT
PO1	Engineering Knowledge	Applies core knowledge of blockchain technology, cryptographic algorithms (RSA, SHA-256), database management (MySQL), and web development (PHP, JavaScript) to build a secure and transparent e-voting system.
PO2	Problem Analysis	Involves analyzing issues in traditional voting systems such as fraud, lack of transparency, and accessibility, and reformulating them into clear technical challenges solved using blockchain and smart contracts.
PO3	Design/Development of Solutions	Focuses on designing and developing a complete blockchain-based voting platform with modules for voter registration, OTP verification, smart contract voting, and real-time result tracking.
PO4	Conduct Investigations of Complex Problems	Involved research (Literature Review) into optimization techniques (Genetic Algorithms, RL) and existing collaboration platforms to inform the design and validate the project approach.
PO5	Modern Tool Usage	Uses modern web technologies, blockchain frameworks (Ethereum, MetaMask), and version control tools (Git) for implementation, testing, and deployment of the system.
PO6	The Engineer and Society	Contributes to the democratic process by providing a secure and tamper-proof voting method that enhances public trust and ensures fair electoral practices.

PO11	Project Management and Finance	Demonstrates effective project planning, resource allocation, and scheduling throughout development. Emphasizes cost-efficient use of tools and technologies to deliver a scalable, secure, and maintainable blockchain-based system within defined constraints.
PO12	Lifelong Learning	Encourages continuous learning and adaptation to emerging technologies such as blockchain, smart contracts, and cybersecurity, fostering a mindset of innovation and self-improvement for future advancements.

LIST OF PROGRAM OUTCOMES (POs):

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

PO2 – Problem Analysis: Identify, formulate, research literature, and analyse 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 systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, 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 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 modeling, to complex engineering activities with an understanding of the limitations.

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

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

PO8 – Ethics : Apply ethical principles and commit to professional ethics and responsibilities and norms of 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 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 – Lifelong Learning: Recognize the need for, and have the ability to engage in independent and life-long learning in the broadest context of technological change.

2. MAPPING WITH PROGRAM SPECIFIC OUTCOMES (PSOs):

SL.NO	PSOs ADDRESSED	RELEVANCE TO PROJECT
1	PSO 1	The project applies advanced computing concepts such as blockchain technology, cryptography, and secure database management (PHP, MySQL) to design a transparent and tamper-proof digital voting system.
2	PSO 2	TrueVote demonstrates the practical use of web and blockchain technologies to solve real-world election challenges by ensuring voter authenticity, data integrity, and election transparency through decentralized digital systems.

LIST OF PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO 1: Apply advanced technologies through innovations to enhance the efficiency of design development.

PSO 2: Apply the principles of computing to analyze, design and implement sustainable solutions for real world challenges.

3. MAPPING WITH SUSTAINABLE DEVELOPMENT GOALS (SDGs):

SDG NO	SDGs ADDRESSED	RELEVANCE TO PROJECT
SDG 9	Industry,Innovation, and Infrastructure	Promotes technological innovation by using blockchain to build a secure, transparent, and efficient digital voting infrastructure.
SDG 10	Reduced Inequality	Ensures equal access to democratic participation by enabling secure and remote voting for all citizens, regardless of location or status.
SDG 11	Sustainable Cities and Communities	Encourages civic engagement and strengthens governance in local and national elections through digital participation.
SDG 16	Peace,Justice,and Strong Institutions	Strengthens democratic institutions by enhancing transparency, accountability, and trust in electoral processes using blockchain technology.
SDG 17	Partnerships for the Goals	Supports collaboration between government, academia, and technology sectors to implement secure, scalable, and inclusive digital governance systems.

SUSTAINABLE DEVELOPMENT GOALS (SDGs):

SDG 1 – No Poverty-End poverty in all its forms everywhere.

SDG 2 – Zero Hunger-End hunger, achieve food security and improved nutrition, and promote sustainable agriculture.

SDG 3 – Good Health and Well-Being-Ensure healthy lives and promote well-being for all at all ages

SDG 4 – Quality Education-Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

SDG 5 – Gender Equality-Achieve gender equality and empower all women and girls.

SDG 6 – Clean Water and Sanitation-Ensure availability and sustainable management of water and sanitation for all.

SDG 7 – Affordable and Clean Energy-Ensure access to affordable, reliable, sustainable, and modern energy for all.

SDG 8 – Decent Work and Economic Growth-Promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all.

SDG 9 – Industry, Innovation, and Infrastructure-Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

SDG 10 – Reduced Inequality-Reduce inequality within and among countries.

SDG 11 – Sustainable Cities and Communities-Make cities and human settlements inclusive, safe, resilient, and sustainable.

SDG 12 – Responsible Consumption and Production-Ensure sustainable consumption and production patterns.

SDG 13 – Climate Action-Take urgent action to combat climate change and its impacts.

SDG 14 – Life Below Water-Conserve and sustainably use the oceans, seas, and marine resources.

SDG 15 – Life on Land -Protect, restore, and promote sustainable use of terrestrial ecosystems, manage forests sustainably, combat desertification, halt and reverse land degradation, and halt biodiversity loss.

SDG 16 – Peace, Justice, and Strong Institutions- Promote peaceful and inclusive societies, provide access to justice for all, and build effective, accountable, and inclusive institutions.

SDG 17 – Partnerships for the Goals -Strengthen the means of implementation and revitalize the global partnership for sustainable develop.

MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR
DEPARTMENT OF COMPUTER APPLICATIONS OCTOBER 2025



DECLARATION

*I hereby certify that the work which is being presented in the project entitled “TRUEVOTE” submitted in the **DEPARTMENT OF COMPUTER APPLICATIONS** is an authentic record of my own work carried under the supervision of **Ms ANJU JOHN, ASSISTANT PROFESSOR**. This study has not been submitted to any other institution or university for the award of any other degree. This report has been checked for plagiarism by the college and the similarity index is within permissible limits set by the college.*

Name & Signature of Student

Date:

Place:

MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR
DEPARTMENT OF COMPUTER APPLICATIONS OCTOBER 2025



CERTIFICATE

*This is to certify that the Project titled “**TRUEVOTE**” is the bonafide record of the work done by **AMAL SHAJI(MLM24MCA-2008)** of Master of Computer Applications towards the partial fulfilment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** by **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**, during the academic year 2025-26.*

Internal Examiner

Project Coordinator

Ms Banu Sumayya S

Assistant Professor

Department of Computer Applications

Project Guide

Head of the Department

Ms. Anju John

Ms. Divya S B

Assistant Professor

Associate Professor

Department of Computer Applications

Department of Computer Applications

ACKNOWLEDGEMENT

I am greatly indebted to the authorities of Mangalam College of Engineering for providing the necessary facilities to successfully complete my Project on the topic “TrueVote”.

I express my sincere thanks to **Dr. Vinodh P Vijayan**, Principal, Mangalam College of Engineering for providing the facilities to complete my Project successfully.

I thank and express my solicit gratitude to **Ms. Divya S B**, HOD, Department of Computer Applications, Mangalam College of Engineering, for her invaluable help and support which helped me a lot in successfully completing this Project work.

I express my gratitude to my Internal Guide, **Ms. Anju John**, Assistant professor, Department of Computer Applications for the suggestions and encouragement which helped in the successful completion of my Project.

Furthermore, I would like to acknowledge with much appreciation the crucial role of the faculties especially Project coordinator, **Ms. Banu Sumayya S**, Department of Computer Applications, Mangalam College of Engineering, who gave the permission to use all the required equipment and the necessary resources to complete the presentation & report.

Finally, I would like to express my heartfelt thanks to my parents who were very supportive both financially and mentally and for their encouragement to achieve my goal.

AMAL SHAJI (MLM24MCA-2008)

ABSTRACT

The increasing demand for secure, transparent, and convenient voting mechanisms has highlighted the limitations of traditional election systems, which often suffer from issues such as tampering, vote duplication, and lack of verifiability. To address these challenges, the TrueVote project introduces a blockchain-based online voting platform that ensures trust, transparency, and data integrity in digital elections.

The system enables voters to cast their votes remotely after OTP-based authentication and Voter ID verification, ensuring that only legitimate users participate. Each vote is recorded as a unique blockchain transaction, making it immutable and publicly verifiable. The platform integrates essential modules including voter registration and authentication, election and candidate management, secure vote casting via smart contracts, and result publication with blockchain verification.

Developed using PHP for backend logic, Solidity for Ethereum smart contracts, and MySQL for structured data storage, the system provides a robust and scalable architecture. The intuitive frontend, built with HTML, CSS, JavaScript, and Bootstrap, offers a seamless user experience for both voters and administrators. By leveraging blockchain's immutability and cryptographic security, TrueVote ensures a tamper-proof, transparent, and trustworthy digital voting process suitable for modern democratic governance.

Mapping with Sustainable Development Goals	Industry,innovation and Infrastructure
SDG 16 Peace, Justice, and Strong Institutions	Promote peaceful and inclusive societies, provide access to justice for all, and build effective, accountable, and inclusive institutions.
SDG 9 Industry, Innovation and Infrastructure	Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	List of Figures	I
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Introduction	2
	1.3 Problem Statement	3
	1.4 Motivation	3
	1.5 Scope	5
2	LITERATURE REVIEW	6
	2.1 A Secure Electronic Voting System Using Blockchain	6
	2.2 Blockchain-Based E-Voting	8
	2.3 Secure Online Voting Using Blockchain with Biometric Integration	10
	2.4 Trustworthy E-Voting Using Blockchain	12
	2.5 Lightweight Blockchain E-Voting for Educational Institutions	13
	2.6 Decentralized E-Voting System Using Ethereum Blockchain	14
	2.7 Blockchain-Enabled E-Voting	15
	2.8 Fair and Transparent Blockchain E-Voting System	16
	2.9 Institutional E-Voting Framework Using Private Blockchain	17
	2.10 Lightweight Blockchain for Secure and Scalable Voting Systems	18
3	PROPOSED SYSTEM	18
	3.1 Blockchain-Based Vote Recording	19
	3.2 Voter Authentication System	19
	3.3 Smart Contract Automation	19
	3.4 One Vote Per User Enforcement	20
	3.5 Election Management	20
	3.6 Candidate Management	20
	3.7 Transparent Result Publishing	20
	3.8 Secure Login and Session Management	20
	3.9 Voting History Dashboard	20

	3.10 Role-Based Access Control	20
	3.11 Data Integrity and Security	21
	3.12 User-Friendly Interface	21
	3.13 Auditable Blockchain Ledger	21
	3.14 Mobile and Web Compatibility	21
4	METHODOLOGY	22
	4.1 Data Collection and User Registration	22
	4.2 Preprocessing and Authentication	23
	4.3 Blockchain Integration and Vote Casting	23
	4.4 Smart Contract Logic and Result Computation	23
	4.5 Security Measures and Data Protection	24
	4.6 System Workflow and Role Interaction	24
	4.7 User Interface and Dashboard Integration	24
5	SYSTEM ARCHITECTURE	26
	5.1 User Interface Layer	27
	5.2 Application Layer	27
	5.3 Database Layer	28
	5.4 Admin Module	28
	5.5 Voter Module	29
	5.6 Data Flow and Communication	29
	5.7 Security and Access Control	29
6	MODULES	31
	6.1 User Management Module	31
	6.2 Authentication and OTP Verification Module	31
	6.3 Election Management Module	32
	6.4 Candidate Management Module	32
	6.5 Voting and Blockchain Integration Module	32
	6.6 Result and Audit Management Module	33
	6.7 Admin Management and Monitoring Module	33
7	DIAGRAMS	34

8	TESTING	41
	8.1 Data collection and preparation	41
	8.2 Model training	41
	8.3 Validation and hyper parameter Tuning	42
	8.4 Integration and User Acceptance Testing	43
	8.5 Security and Reliability Testing	44
	8.6 Deployment and Monitoring	44
9	ADVANTAGES & DISADVANTAGES	46
	9.1 Advantages	46
	9.2 Disadvantages	46
10	RESULTS	48
11	CONCLUSION & FUTURE SCOPE	54
	APPENDICES	56
	REFERENCES	59

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	SYSTEM ARCHITECTURE	26
7.1	ER DIAGRAM	34
7.2	LEVEL-0 DFD	35
7.3	LEVEL 1 DFD (VOTER)	36
7.4	LEVEL 1 DFD (ADMIN)	37
7.5	USE CASE DIAGRAM	38
7.6	CLASS DIGRAM	39
7.7	SEQUENCE DIAGRAM	40
10.1	LOGIN INTERFACE	48
10.2	USER DASHBOARD INTERFACE	49
10.3	ELECTION PAGE	50
10.4	VOTING PAGE	51
10.5	RESULT PAGE	52
10.6	ADMIN DASHBOARD INTERFACE	53

LIST OF ABBREVIATIONS

ABBREVIATION		FULL FORM
OTP	-	One-Time Password
ID	-	Identifier (Voter ID)
PoW	-	Proof of Work
AI	-	Artificial Intelligence
PoS	-	Proof of Stake
PBFT	-	Practical Byzantine Fault Tolerance
PoA	-	Proof of Authority
SHA-256	-	Secure Hash Algorithm 256-bit
PHP	-	Hypertext Preprocessor
MySQL	-	Relational Database Management System

CHAPTER 1

INDRODUCTION

1.1 Background

Elections are the foundation of any democratic system, providing citizens with the right to choose their representatives and influence governance. However, traditional voting systems—whether paper-based or electronic—continue to face significant challenges related to security, transparency, accessibility, and efficiency. Issues such as vote tampering, duplicate voting, human error, and lack of verifiable audit trails have often raised concerns about the credibility and integrity of election outcomes. In many cases, logistical constraints and limited accessibility have also discouraged voter participation, especially among citizens residing in remote or overseas locations.

With the rapid advancement of digital technologies, there is a growing demand for secure, transparent, and convenient online voting systems. While online voting offers the potential to simplify the election process and increase participation, it also introduces new challenges related to data privacy, authentication, and system trustworthiness. A secure framework is essential to ensure that each vote is cast by a verified voter, counted accurately, and cannot be altered or deleted once submitted.

Blockchain technology has emerged as a revolutionary solution to address these challenges. Its decentralized and immutable nature ensures that every transaction—representing a vote—is securely recorded in a distributed ledger that cannot be manipulated. This guarantees transparency, verifiability, and data integrity, eliminating the risk of unauthorized modifications. By integrating blockchain with robust authentication mechanisms such as Voter ID verification and OTP-based login, it becomes possible to design a tamper-proof and trustworthy digital voting platform.

The TrueVote project was conceptualized to fulfill this need for a secure, transparent, and user-friendly online voting system. It bridges the gap between traditional voting limitations and modern digital security standards. The platform allows voters to register securely, verify their identity, and cast votes from any location, with each vote being permanently stored on the Ethereum blockchain as a unique transaction. The system includes dedicated modules

for admins **to** manage elections, candidates, and voter approvals, while voters can participate in elections, track their vote history, and view real-time results.

Ultimately, TrueVote aims to strengthen the democratic process by ensuring that every vote counts and remains immutable. By leveraging blockchain technology and secure web mechanisms, it promotes a transparent, efficient, and tamper-proof voting environment, suitable for modern digital governance. The system not only enhances election integrity but also encourages greater voter participation, accessibility, and trust in the electoral process.

1.2 Introduction

Elections are a vital component of democratic governance, allowing citizens to express their choices and influence decision-making at various levels of administration. However, traditional voting systems continue to face several challenges such as vote tampering, lack of transparency, logistical inefficiencies, and limited accessibility for remote or disabled voters. Manual vote counting and centralized data handling increase the risk of manipulation, while long queues and restricted polling locations often discourage participation. These issues underline the urgent need for a secure, transparent, and accessible alternative to conventional voting methods.

In recent years, digital technology has transformed numerous aspects of civic and administrative processes, paving the way for electronic voting (e-voting) solutions. However, most existing online voting systems still depend on centralized databases, which remain vulnerable to hacking, unauthorized alterations, and data loss. There is, therefore, a strong need for a system that ensures tamper-proof vote recording, voter authentication, and result verification through a decentralized and transparent framework.

The TrueVote project aims to bridge this gap by developing a blockchain-based online voting platform that ensures integrity, trust, and security in digital elections. The system enables voters to register using their Voter ID, complete authentication through OTP verification, and cast their vote securely via a smart contract deployed on the Ethereum blockchain. Each vote is stored as a unique, immutable blockchain transaction, ensuring that no entity can modify or delete it once recorded. Administrators can create elections, verify voter registrations, manage candidates, and publish results, while voters can view elections, cast votes, and check their voting history — all within a secure and user-friendly interface.

Through this project, we aim to strengthen democratic participation by leveraging blockchain technology to make elections transparent, verifiable, and tamper-proof. The TrueVote system represents a step toward digital democracy, where technology empowers citizens to vote securely and confidently from anywhere, ensuring that every vote is counted and every voice truly matters.

1.3 Problem Statement

The process of conducting elections plays a crucial role in maintaining democracy, yet traditional voting systems continue to face several persistent challenges. Conventional methods, such as paper ballots and electronic voting machines, are often vulnerable to issues like vote tampering, human error, duplicate voting, and lack of verifiable transparency. These problems raise doubts about the credibility and fairness of election outcomes. Additionally, the manual handling of votes and centralized data storage make traditional systems susceptible to manipulation and data breaches.

In many regions, logistical barriers such as long queues, limited polling stations, and inconvenient voting hours discourage voter participation. Remote and overseas voters often find it difficult to access the voting process, leading to reduced voter turnout. Although some online voting platforms have been developed, most rely on centralized databases that can be compromised through cyberattacks or unauthorized access, posing serious risks to election integrity.

Therefore, the main problem addressed by this project is the lack of a secure, transparent, and decentralized voting platform that ensures voter authenticity, prevents vote duplication, and provides publicly verifiable results. The system should allow eligible voters to cast their votes conveniently from any location while guaranteeing data privacy and immutability. The TrueVote project aims to overcome these limitations by developing a blockchain-based voting system that enhances security, transparency, and trust in the electoral process.

1.4 Motivation

In today's digital era, elections remain one of the most critical components of a democratic society, yet the process of conducting them securely and transparently continues to pose significant challenges. The increasing cases of vote manipulation, data breaches, and lack of verifiable transparency in traditional and electronic voting systems have raised serious

concerns about the integrity of election outcomes. Observing these issues in real-world elections motivated the idea of creating a system that could ensure secure, verifiable, and tamper-proof voting through advanced technology.

The motivation behind developing the TrueVote system stems from the vision of transforming the voting process into a more transparent, accessible, and trustworthy activity. Instead of relying on centralized servers or manual vote counting, this system encourages the use of blockchain technology to record votes immutably and verifiably. Blockchain ensures that once a vote is cast, it cannot be altered or deleted, thereby maintaining complete trust in the system. Additionally, it simplifies the voting process for citizens, allowing them to participate securely from anywhere, ensuring accessibility and convenience while maintaining data integrity and voter privacy.

From a technical and educational perspective, the project also serves as an excellent opportunity to apply core concepts of modern web and blockchain development. It involves implementing database management, secure authentication, smart contracts, and encrypted transactions in a real-world context. Building such a system enhances practical skills in both backend and frontend development while also addressing a genuine societal need. The TrueVote project demonstrates how technology can be effectively used to strengthen democratic systems and promote digital trust.

Furthermore, the project is motivated by the need for transparency, fairness, and accountability in the electoral process. Many people lose confidence in elections due to the lack of reliable systems that ensure vote security and public verifiability. By integrating features like voter registration, OTP-based authentication, and blockchain-based vote recording, this system provides a secure and accountable environment for both administrators and voters.

Ultimately, this project is driven by the goal of developing an innovative, user-friendly, and socially beneficial application. The TrueVote system not only aims to enhance election transparency and security but also aspires to bring a positive change in democratic participation. It reflects a collective step toward secure digital governance — promoting electoral integrity, citizen empowerment, and technological advancement in the voting process.

1.5 Scope

The TrueVote system is designed to function as a comprehensive web-based platform that facilitates secure, transparent, and efficient online voting using blockchain technology. The scope of the project includes voter registration and authentication, election creation and management, candidate registration and approval, OTP-based vote casting, and blockchain transaction recording for verifiable results.

Voters can register on the platform using their Voter ID and complete verification through an OTP sent to their registered email. Once verified, voters can view active or upcoming elections, check candidate details, and cast their vote securely. Each vote is recorded as a unique transaction on the Ethereum blockchain, ensuring that it cannot be altered or deleted after submission. Voters can also view their voting history and check published results once elections are completed.

The admin module oversees the overall system operations, including voter approval, election scheduling, candidate management, and result publication. It also monitors blockchain transaction hashes to verify the authenticity and integrity of recorded votes. However, the system does not include features such as biometric authentication or live election analytics, as its focus is on providing a secure, decentralized, and verifiable voting process rather than real-time monitoring.

In summary, the scope of this project covers the development of a secure and transparent blockchain-based voting system that simplifies election management while maintaining high levels of data integrity and trust. The project emphasizes usability, authenticity, and the promotion of digital democracy through innovative and reliable technology.

CHAPTER 2

LITERATURE REVIEW

2.1 A Secure Electronic Voting System Using Blockchain Technology [Alzahrani, Bulusu, and Shiva (IEEE Access)]

This research paper introduces a secure and transparent electronic voting system built on blockchain technology. The main goal of the study is to overcome the limitations of traditional voting methods, such as lack of transparency, tampering risks, and dependence on centralized authorities. The system ensures trust, integrity, and verifiability in the voting process through decentralized ledger technology.

The authors emphasize that blockchain can provide a distributed, immutable, and auditable record of votes, making manipulation or unauthorized modification virtually impossible. The system uses cryptographic mechanisms and smart contracts to automate vote recording and result tallying while maintaining voter privacy and system transparency.

Objectives

- To design a decentralized and tamper-proof voting system using blockchain.
- To ensure voter anonymity, integrity of votes, and public verifiability.
- To reduce reliance on centralized authorities in managing election data.
- To increase trust and transparency in the election process.

Methodology

The general methodology involves:

- **Voter Registration:** Each eligible voter is registered through a secure authentication process, and their identity is verified by an authorized authority. After verification, a unique digital ID is created.
- **Blockchain Setup:** A private or permissioned blockchain network is established, ensuring only authenticated entities (like election commissions or nodes) can participate in consensus and data validation.

- **Smart Contracts:** Smart contracts are used to handle election logic automatically — including candidate information storage, vote casting, and result calculation — ensuring transparency and eliminating manual interference.
- **Vote Casting:** Each voter casts their vote through a blockchain-enabled web interface or application. The vote is encrypted and recorded as a transaction on the blockchain, ensuring it cannot be modified or duplicated.
- **Vote Counting and Verification:** Votes are automatically tallied through the blockchain's consensus mechanism. The results can be verified by anyone, ensuring auditability and transparency.
- **Security and Privacy:** The system uses public-key cryptography and hashing to secure the voting process. It ensures that while votes are transparent on the blockchain, voter identities remain anonymous.

Key Findings

The authors conclude that blockchain-based e-voting significantly enhances election security, transparency, and public trust. The system:

- Prevents vote tampering or duplication through immutability.
- Maintains voter anonymity through cryptographic encryption.
- Provides a verifiable and auditable election record.
- Eliminates the need for intermediaries or centralized control.

The study also identifies challenges, such as scalability, voter device security, and network latency, as future research areas. Overall, the proposed system demonstrates that blockchain can revolutionize democratic voting by creating a secure, transparent, and tamper-proof electoral process.

2.2 Blockchain-Based E-Voting: A Review of Technologies and Implementation Models [P. S. Praveen Kumar et al. (Elsevier)]

This paper provides a detailed review of various blockchain-based electronic voting systems and the technologies that power them. The authors examine how blockchain can transform traditional voting mechanisms by ensuring transparency, immutability, and decentralization. They analyze multiple implementation models, consensus mechanisms, and cryptographic techniques used in recent research to enhance election integrity and voter trust.

The paper highlights the importance of blockchain's distributed ledger in eliminating single points of failure and preventing vote tampering. It also discusses how smart contracts can automate different stages of the voting process, such as voter registration, vote casting, and result tabulation, without human intervention.

Objectives

- To review the existing blockchain-based e-voting systems and their architectures.
- To analyse the role of different blockchain platforms and consensus algorithms in e-voting.
- To identify key challenges and limitations in the deployment of blockchain voting systems.
- To propose improvements for achieving scalability, privacy, and usability in digital elections.

Methodology

The study uses a comparative review approach, analysing multiple blockchain-based e-voting prototypes and models proposed in previous literature. The authors evaluate systems based on factors such as technology stack, security mechanisms, transaction handling, and performance.

1.Blockchain Platforms Reviewed: The paper reviews blockchain frameworks like Ethereum, Hyperledger Fabric, and EOS, comparing their suitability for voting applications in terms of speed, cost, and scalability.

2.Consensus Mechanisms: The study discusses consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), emphasizing that PBFT and PoS are more efficient and suitable for permissioned election networks.

3.Security and Cryptography: The authors highlight the use of public-key cryptography, hashing, and zero-knowledge proofs to maintain vote secrecy and prevent voter identification or vote manipulation.

4.Smart Contract Integration: The paper explains how smart contracts manage and automate voting procedures—verifying voter eligibility, ensuring one vote per person, and performing transparent result aggregation.

5.Evaluation Criteria: Each model is assessed for security, transparency, anonymity, scalability, and system efficiency to identify strengths and areas for improvement.

Key Findings

The authors conclude that blockchain-based voting systems offer a promising solution to major issues in conventional elections such as fraud, lack of transparency, and centralized control. They observe that:

- Decentralization eliminates the risk of vote tampering.
- Cryptographic techniques preserve voter privacy.
- Smart contracts enhance automation and transparency.
- Public verification builds trust among voters and authorities.

However, the paper also notes limitations in **scalability**, **energy efficiency**, and **user accessibility**. It recommends future research into lightweight consensus protocols and hybrid blockchain architectures to make e-voting more practical and widely adoptable.

Overall, the study provides a strong foundation for developing secure, efficient, and transparent e-voting systems by leveraging blockchain’s core features of decentralization, immutability, and cryptographic trust.

2.3. Secure Online Voting Using Blockchain with Biometric Integration [Frag, A., Mousa, M., El-Bakry, H.]

This paper introduces a secure online voting system that integrates blockchain technology with biometric authentication to ensure election transparency, voter identity verification, and data integrity. The authors aim to address the key challenges in electronic voting systems such as identity fraud, vote duplication, and centralized data control by combining biometric verification with decentralized blockchain-based storage.

The proposed system provides a secure framework that enables voters to authenticate themselves through fingerprint or facial recognition before casting their votes. By using

blockchain as a distributed ledger, all voting records are stored immutably, ensuring that votes cannot be altered, deleted, or forged

Objectives

- To develop a secure and transparent online voting model using blockchain technology.
- To integrate biometric authentication for accurate voter verification and fraud prevention.
- To ensure data integrity, privacy, and immutability of votes.
- To enhance the reliability and public trust in electronic voting systems.

Methodology

The authors design a multi-layered architecture combining biometric systems, blockchain infrastructure, and cryptographic protocols to secure every stage of the voting process.

1.User Registration: Each voter registers with personal and biometric details (such as fingerprint or facial data), which are verified and stored in an encrypted format within the system.

2.Biometric Authentication: Before voting, the user's biometric input is re-verified to confirm identity, ensuring that only legitimate voters can access the voting interface.

3.Blockchain Layer: Once authentication is completed, the voting transaction is recorded on a blockchain ledger. Each vote is represented as a unique encrypted transaction, preventing any modification or duplication.

4.Encryption and Hashing: Cryptographic hashing techniques (such as SHA-256) are used to secure the vote data, ensuring that the content of the votes remains confidential and tamper-proof.

5.Consensus Mechanism: A permissioned blockchain model is employed, often using Practical Byzantine Fault Tolerance (PBFT) or Proof of Authority (PoA) for efficient validation without high computational cost.

6. Vote Counting and Verification: The counting process is automated through smart contracts, ensuring transparency and accuracy. Results can be audited by verifying blockchain transactions.

Key Findings

The study concludes that integrating biometric authentication with blockchain significantly enhances the security, transparency, and trustworthiness of online voting systems. The system ensures that:

- Only registered and authenticated voters can cast a vote.
- Each vote is securely recorded and immutable.
- Voter anonymity is preserved while maintaining verifiability.
- Centralized manipulation or data breaches are effectively prevented.

The authors also highlight challenges such as the cost of biometric infrastructure, data storage requirements, and network scalability, which need further optimization. Future enhancements may include using lightweight blockchain frameworks and advanced encryption schemes to support large-scale elections.

Overall, the paper demonstrates that combining blockchain and biometrics provides a robust and secure foundation for next-generation electronic voting systems, offering both accountability and convenience in digital democracy.

2.4 Trustworthy E-Voting Using Blockchain: Challenges and Future Scope [Hossain, M. S. et al. (Springer)]

Modern democratic systems face significant challenges in maintaining transparency, trust, and security in electronic voting (e-voting) processes. Traditional centralized voting systems are vulnerable to data manipulation, cyberattacks, and lack of public verifiability. Hossain et al. (Springer) propose a blockchain-based e-voting framework to establish a trustworthy, transparent, and tamper-proof voting system that enhances voter confidence while ensuring privacy and fairness. The study explores how blockchain's decentralized architecture, combined with cryptographic techniques, can revolutionize digital voting by eliminating third-party dependency and securing election data.

The paper emphasizes the importance of immutability, voter anonymity, and public verifiability in e-voting systems. By recording votes as transactions on a distributed ledger, blockchain ensures that once a vote is cast, it cannot be altered or deleted. This creates an auditable and transparent voting process where all stakeholders can verify the legitimacy of results without compromising voter privacy.

Key Aspects:

- **Blockchain-Based Architecture:** The proposed system uses a permissioned blockchain where registered authorities act as validating nodes. Each vote is treated as a unique transaction stored permanently on the distributed ledger. This decentralization eliminates single points of failure and ensures data integrity throughout the election process.
- **Voter Authentication and Privacy:** Voters are authenticated using unique digital credentials, ensuring that only eligible participants can cast a vote. Cryptographic methods such as public-key encryption and hash functions protect voter identities while maintaining the anonymity of ballots.
- **Smart Contract Automation:** The system leverages **smart contracts** to automatically execute election operations, including vote recording, counting, and result publishing. This automation minimizes human intervention and prevents tampering or vote duplication.
- **Transparency and Verifiability:** Every vote recorded on the blockchain can be publicly verified without revealing voter information. The immutable nature of blockchain ensures that all voting data remains auditable and transparent for independent validation.
- **Security and Data Integrity:** Blockchain's cryptographic foundations protect against unauthorized modifications and double voting. The consensus mechanism, such as Proof of Authority (PoA) or Practical Byzantine Fault Tolerance (PBFT), ensures trust among validating nodes while maintaining high transaction throughput.
- **Challenges Identified:** The study outlines potential limitations, including scalability issues in handling national-level elections, high computational overhead, and the need for comprehensive legal and regulatory frameworks to support blockchain-based voting.
- **Future Scope:** The authors suggest exploring AI-driven anomaly detection, zero-knowledge proofs, and hybrid blockchain models to enhance system scalability and security.

Integration with biometric verification and mobile-based voting interfaces is also recommended for improved accessibility and user trust.

2.5 Lightweight Blockchain E-Voting for Educational Institutions [Singh, A., Kumar, S. (IJCSNS)]

Educational institutions require secure, transparent, and easily auditable voting systems for student elections and other organizational decisions. Traditional paper-based or centralized electronic voting systems are prone to manipulation, human errors, and lack transparency. Singh and Kumar (IJCSNS) propose a lightweight blockchain-based e-voting system specifically designed for educational environments, aiming to provide a secure, verifiable, and efficient voting mechanism while minimizing computational and infrastructural overhead.

The paper highlights how blockchain's decentralized ledger, combined with lightweight cryptographic techniques, can simplify the election process for small-scale environments such as universities and colleges. By recording votes as blockchain transactions, the system ensures immutability, voter privacy, and verifiability, even in resource-constrained settings.

Key Aspects:

- **Lightweight Blockchain Architecture:** The system uses a permissioned blockchain with minimal computational requirements, suitable for educational institutions. Voting authorities act as validating nodes, ensuring vote integrity while maintaining low resource consumption.
- **Voter Authentication and Anonymity:** Students are authenticated using unique digital credentials issued by the institution. Cryptographic hashing and encryption protect voter identities, ensuring that votes remain anonymous yet verifiable.
- **Smart Contract-Based Voting:** Smart contracts automatically manage voting operations, including vote submission, validation, counting, and result publication. This reduces manual effort and prevents double voting or manipulation.
- **Transparency and Auditability:** All votes are recorded on the blockchain, allowing administrators and stakeholders to audit the election results without compromising voter anonymity.

- **Security and Integrity:** The system leverages lightweight consensus mechanisms suitable for small-scale deployments, ensuring secure vote recording, tamper-resistance, and protection against unauthorized modifications.
- **Challenges Identified:** The study notes potential limitations such as network latency, adoption resistance among students, and the need for secure digital credential management.
- **Future Scope:** The authors suggest integrating mobile voting interfaces, biometric verification, and scalable consensus algorithms to enhance accessibility, usability, and security. Further research on hybrid lightweight blockchain models could enable broader adoption in larger educational networks.

2.6 Decentralized E-Voting System Using Ethereum Blockchain [Ali, R., Anwar, Z., & Khan, S. (IEEE Xplore)]

Ali and colleagues proposed a decentralized Ethereum-based e-voting framework that ensures transparency, immutability, and voter anonymity. The system leverages smart contracts to automate vote casting and tallying, thereby eliminating the need for third-party supervision. Although effective in preventing tampering, the study noted high gas costs and transaction delays, making it less practical for smaller organizations.

Key Aspects:

- **Smart Contract Automation:** Enables automatic vote recording and counting without human interference.
- **Decentralized Ledger:** Ensures immutability and tamper-proof record storage across multiple nodes.
- **Voter Anonymity:** Achieved through cryptographic techniques that separate identity from the vote.
- **Transparency and Verifiability:** Every transaction is traceable on the blockchain, ensuring public auditability.
- **Challenges:** High transaction fees (gas costs), latency in Ethereum network, and scalability concerns for institutional use.
- **Relevance:** Highlights the need for lightweight blockchain systems like Singh & Kumar's, optimized for resource-limited educational environments.

2.7. Blockchain-Enabled E-Voting [Kshetri, N., & Voas, J. (IEEE Computer)]

Kshetri and Voas explored the potential of blockchain technology in transforming traditional voting systems by providing a secure, transparent, and tamper-resistant framework. Their study emphasized how decentralized trust mechanisms can address election fraud, vote manipulation, and transparency issues. However, they also discussed practical challenges such as scalability, identity management, and voter accessibility that hinder mass adoption.

Key Aspects:

- **Decentralized Trust Model:** Removes the need for central authorities in vote storage and verification.
- **Immutable Ledger:** Guarantees that votes cannot be altered once recorded.
- **Enhanced Transparency:** Publicly verifiable blockchain records increase trust in election outcomes.
- **Security Concerns:** Addresses risks of tampering and unauthorized access through consensus mechanisms.
- **Challenges:** Scalability limitations, lack of voter identity verification standards, and usability barriers.
- **Relevance:** Provides a theoretical foundation for adopting blockchain in voting systems, inspiring lightweight, scalable models like Singh & Kumar's designed for educational institutions.

2.8 Fair and Transparent Blockchain E-Voting System [Hardwick, F. S., Akram, R. N., & Markantonakis, K. (IEEE Transactions on Emerging Topics in Computing)]

Hardwick and colleagues developed a fair and transparent e-voting protocol based on blockchain that emphasizes end-to-end verifiability and auditability. Their design ensures that each vote is recorded immutably while maintaining voter privacy and preventing double voting. The study also explored consensus mechanisms for small-scale deployments and proposed improvements to reduce latency and increase efficiency.

Key Aspects:

- **End-to-End Verifiability:** Voters can verify that their votes are correctly recorded without revealing identities.

- Immutable Blockchain Ledger: Prevents vote alteration and enables public auditing.
- Anti-Double Voting Mechanism: Smart contract logic ensures each voter can vote only once.
- Consensus Efficiency: Evaluates low-latency algorithms suitable for smaller environments.
- Challenges: Balancing transparency with voter privacy and managing blockchain scalability.
- Relevance: Supports the idea of secure, efficient, and auditable voting systems, aligning with Singh & Kumar's goal of implementing low-resource, lightweight blockchain-based e-voting for academic institutions.

2.9 Institutional E-Voting Framework Using Private Blockchain [Patel, V., & Shah, K. (IJACSA)]

Patel and Shah introduced an e-voting framework tailored for academic institutions using a private permissioned blockchain. The system employed hashed student identifiers for voter authentication and institutional nodes for vote validation. It was designed to improve trust, transparency, and accountability in student elections but required technical expertise for setup and maintenance, limiting ease of deployment.

Key Aspects:

- Permissioned Blockchain: Restricted validator access to institutional authorities for controlled participation.
- Hashed Voter IDs: Ensured anonymity while allowing verification of vote legitimacy.
- Distributed Validation: Reduced chances of single-point failure or data manipulation.
- Auditability: All voting records were available for transparent verification post-election.
- Challenges: Technical complexity in deployment and dependency on skilled personnel.
- Relevance: Shares the same institutional application domain as Singh & Kumar's study but emphasizes the need for simpler and more lightweight blockchain designs suited for real-world educational environments.

2.10 Lightweight Blockchain for Secure and Scalable Voting Systems [Noor, T. H., Zeadally, S., & Alfazi, A. (Elsevier)]

Noor and colleagues proposed a lightweight blockchain-based voting model designed to enhance scalability, speed, and energy efficiency. The system utilizes a Proof of Authority (PoA) consensus mechanism and efficient cryptographic algorithms to maintain security while minimizing computational cost. The authors also explored mobile and web-based interfaces to improve voter accessibility in local and institutional elections.

Key Aspects:

- **Lightweight Architecture:** Optimized for low-resource environments with minimal hardware requirements.
- **Proof of Authority (PoA) Consensus:** Ensures fast block confirmation and reduced energy consumption.
- **Secure Encryption:** Protects voter data and ballot privacy using lightweight cryptographic methods.
- **Cross-Platform Support:** Enables participation via mobile and web applications.
- **Challenges:** Balancing performance and decentralization while maintaining security.
- **Relevance:** Closely aligns with Singh & Kumar's approach, providing strong evidence that lightweight blockchain models are ideal for educational institutions needing secure yet resource-efficient e-voting solutions.

CHAPTER 3

PROPOSED SYSTEM

The proposed TrueVote system is designed to provide a secure, transparent, and user-friendly platform for conducting online elections using advanced blockchain technology.

It aims to eliminate electoral fraud, manipulation, and unauthorized access, ensuring that every vote is cast and recorded accurately while maintaining the integrity of the election process.

Unlike conventional electronic voting systems that rely on centralized servers susceptible to hacking or tampering, TrueVote leverages a decentralized blockchain ledger, making each vote immutable, traceable, and auditable.

Voter authentication is strengthened using a dual verification mechanism consisting of Voter ID validation and OTP (One-Time Password) authentication, ensuring that only eligible participants can cast votes.

The system employs smart contracts to automate the entire voting workflow, including vote recording, vote counting, and result publication, significantly minimizing human errors, intervention, and the possibility of duplicate voting.

Each voter has access to a personal voting history dashboard, enabling them to verify that their vote has been successfully recorded and to review results from past elections.

Administrators are provided with a robust dashboard to manage elections efficiently, including the creation of elections, management of candidates, scheduling of election dates, monitoring of blockchain transactions, and secure publication of results.

TrueVote ensures that all sensitive data, including voter credentials and election details, are encrypted and securely stored in a structured database integrated with the blockchain for redundancy and verification. The system supports role-based access control, allowing

separate functionalities for voters and administrators to prevent unauthorized operations and maintain the security and integrity of the platform.

By combining cryptographic security, blockchain immutability, automated smart contract operations, and reliable authentication mechanisms, TrueVote guarantees fair, transparent, and verifiable elections.

The platform is also designed to be scalable, accessible, and adaptable for different election sizes, from small institutional polls to larger organizational or regional elections, without compromising speed or security.

Additional features such as real-time notifications, audit trails, secure session management, and encrypted storage of voter information further enhance trust, convenience, and reliability.

Overall, TrueVote represents a comprehensive solution for digital elections, ensuring transparency, accountability, and confidence among all stakeholders while providing a smooth and intuitive voting experience.

3.1 Blockchain-Based Vote Recording

Each vote is treated as a unique transaction on the blockchain, ensuring immutability and traceability. Votes cannot be altered, deleted, or duplicated, providing full transparency and trustworthiness.

3.2 Voter Authentication System

Voters must verify their Voter ID and OTP before accessing the system. This prevents unauthorized access and ensures that only eligible voters participate.

3.3 Smart Contract Automation

Smart contracts automatically execute election operations such as vote submission, tallying, and result declaration. This eliminates human intervention and prevents errors or manipulation.

3.4 One Vote Per User Enforcement

The system ensures single-vote enforcement, so no voter can vote more than once per election. Smart contracts verify voter eligibility in real time during vote submission.

3.5 Election Management

Admins can create elections with details such as election code, title, type, constituency, start date, and end date. Election status is updated dynamically as Upcoming, Ongoing, or Completed.

3.6 Candidate Management

Admins can add candidates with party name, symbol, manifesto, and verification documents. Candidate details are visible to voters to make informed voting decisions.

3.7 Transparent Result Publishing

Results are published through the blockchain, ensuring that votes are auditable and verifiable by both administrators and voters. Any disputes can be resolved by reviewing transaction records.

3.8 Secure Login and Session Management

The platform uses OTP-based login, encrypted passwords (bcrypt/PHP password_hash), and automatic session expiry. This prevents unauthorized access and protects voter accounts.

3.9 Voting History Dashboard

Voters can track their voting history, submitted votes, and election results. This enhances auditability and trust in the system.

3.10 Role-Based Access Control

Admins and voters have distinct permissions, preventing unauthorized actions. Only admins can create elections, approve candidates, and monitor blockchain transactions, while voters can cast votes and view results.

3.11 Data Integrity and Security

Blockchain cryptography ensures that vote data is secure, immutable, and tamper-proof. MySQL database stores voter profiles, election details, and candidate information securely.

3.12 User-Friendly Interface

The platform is designed to be intuitive and easy to navigate for both technical and non-technical users.

Clear interfaces for admin dashboards and voter dashboards improve usability and engagement.

3.13 Auditable Blockchain Ledger

Every vote has a unique transaction hash, which can be used to verify authenticity without compromising voter privacy. This creates trust and accountability in the election process.

3.14 Mobile and Web Compatibility

TrueVote supports both desktop and mobile devices, enabling users to participate from anywhere. The responsive design ensures a smooth voting experience across platforms.

3.15 Secure Result Backup

All election data and results are stored on the blockchain and mirrored in a secure database backup, preventing data loss.

CHAPTER 4

METHODOLOGY

The methodology for the TrueVote project follows a structured and multi-phase approach aimed at developing a secure, transparent, and decentralized online voting system using blockchain technology. It integrates PHP-based web development for the application layer, MySQL for efficient data management, and Ethereum smart contracts for blockchain integration. The goal is to ensure that each vote remains authentic, immutable, and verifiable while maintaining user privacy and the overall integrity of the election process.

This approach is designed to streamline the entire election workflow—from voter registration to result declaration—while ensuring data accuracy, user trust, and system scalability. The methodology progresses through several stages, including voter registration, authentication, election setup, blockchain-based vote casting, and result verification. Each component is carefully coordinated to maintain a tamper-proof and verifiable voting experience. Overall, this methodology ensures that both voters and administrators interact within a secure, transparent, and user-friendly digital environment that upholds democratic principles and fosters trust in digital voting.

4.1 Data Collection and User Registration

The first phase of the TrueVote system involves collecting essential voter and election-related data. During registration, voters provide basic information such as their name, email, Voter ID, and password. Administrators then verify and approve these registrations after validating the Voter ID details to ensure authenticity. Meanwhile, candidate and election information—such as the election title, code, type, and constituency—are entered by the admin through dedicated web forms.

All the collected data is stored in relational MySQL tables such as voters, elections, candidates, and votes, which are linked using foreign key relationships to maintain referential integrity. Data validation mechanisms prevent duplicate or invalid registrations, while sensitive information such as passwords remains encrypted to ensure voter privacy and data protection.

4.2 Preprocessing and Authentication

Before the voting phase begins, the system undergoes preprocessing and authentication to ensure that only eligible voters are allowed to participate. Each user must verify their identity through an email-based OTP system, which prevents impersonation and unauthorized access. During this phase, session validation mechanisms are also enforced to restrict voters to a single active session, thereby preventing multiple logins from different devices.

Additionally, an eligibility check is performed to ensure that voters can participate only in the elections for which they are registered and that are currently active. This preprocessing phase guarantees the accuracy and authenticity of voter and election data before any interaction with the blockchain layer takes place.

4.3 Blockchain Integration and Vote Casting

The core of the TrueVote system lies in its integration with blockchain technology, which ensures transparency, immutability, and accountability. Once a voter's identity is successfully verified through the OTP process, they are permitted to cast their vote via the web interface. The selected candidate's ID, along with the voter's unique details, is transmitted to a smart contract deployed on the Ethereum blockchain.

The smart contract validates whether the voter has already voted in that particular election, ensuring the "one person, one vote" principle. Each submitted vote is stored as a unique blockchain transaction, identified by a distinct transaction hash. This hash makes every vote publicly verifiable and immutable, guaranteeing that no one can alter or delete recorded votes. Through this mechanism, the TrueVote system eliminates the possibility of vote manipulation or tampering, thereby ensuring complete transparency.

4.4 Smart Contract Logic and Result Computation

Smart contracts, written in Solidity, manage the backend logic for recording and counting votes. Once a transaction is confirmed on the blockchain, the corresponding candidate's vote tally is automatically updated. These smart contracts also contain access control features that allow only administrators to create or close elections, ensuring that sensitive operations are restricted to authorized personnel.

At the conclusion of an election, the results are computed automatically on the blockchain and displayed on both the admin and voter dashboards. This automation removes the need for manual vote counting, significantly reducing the chances of human error and enhancing public trust in the election process through transparent result computation.

4.5 Security Measures and Data Protection

Security plays a crucial role at every stage of the TrueVote system. User passwords are encrypted using PHP's `password_hash()` function before being stored in the database, ensuring that sensitive credentials remain protected. Session management is implemented to prevent simultaneous multiple logins by a single user, and blockchain data remains unalterable once recorded, guaranteeing complete immutability.

Additionally, regular database backups and strict server-side validations are performed to safeguard against data corruption and unauthorized access. Administrative permissions are also restricted to ensure that only verified users can perform critical operations such as approving candidates or publishing results. These combined security measures maintain the integrity and reliability of both the centralized and decentralized components of the system.

4.6 System Workflow and Role Interaction

The overall workflow of the TrueVote platform integrates three primary modules: Admin, Voter, and Blockchain Layer. The administrator is responsible for creating and managing elections, verifying voter and candidate information, and publishing results. The voter, on the other hand, registers on the platform, verifies their identity through an OTP, views available elections, casts their vote via blockchain, and later checks the published results.

Meanwhile, the blockchain layer executes smart contract functions, validates transactions, and immutably stores vote hashes. This modular interaction ensures seamless coordination between all system components, maintaining a transparent and efficient election lifecycle from registration to result verification.

4.7 User Interface and Dashboard Integration

The user interface of TrueVote is designed using HTML, CSS, JavaScript, and Bootstrap, providing an intuitive and responsive experience for both voters and administrators. The voter dashboard displays relevant election details, candidate profiles, and a record of voting

history, including blockchain transaction hashes for transparency. The administrator dashboard offers comprehensive control over voter management, candidate verification, election scheduling, and the publication of results.

Through its interactive design and responsive layout, the interface ensures accessibility across different devices. Additionally, real-time notifications and feedback messages keep users informed about important election updates, verification statuses, and results, contributing to an engaging and transparent digital voting experience.

CHAPTER 5

SYSTEM ARCHITECHURE

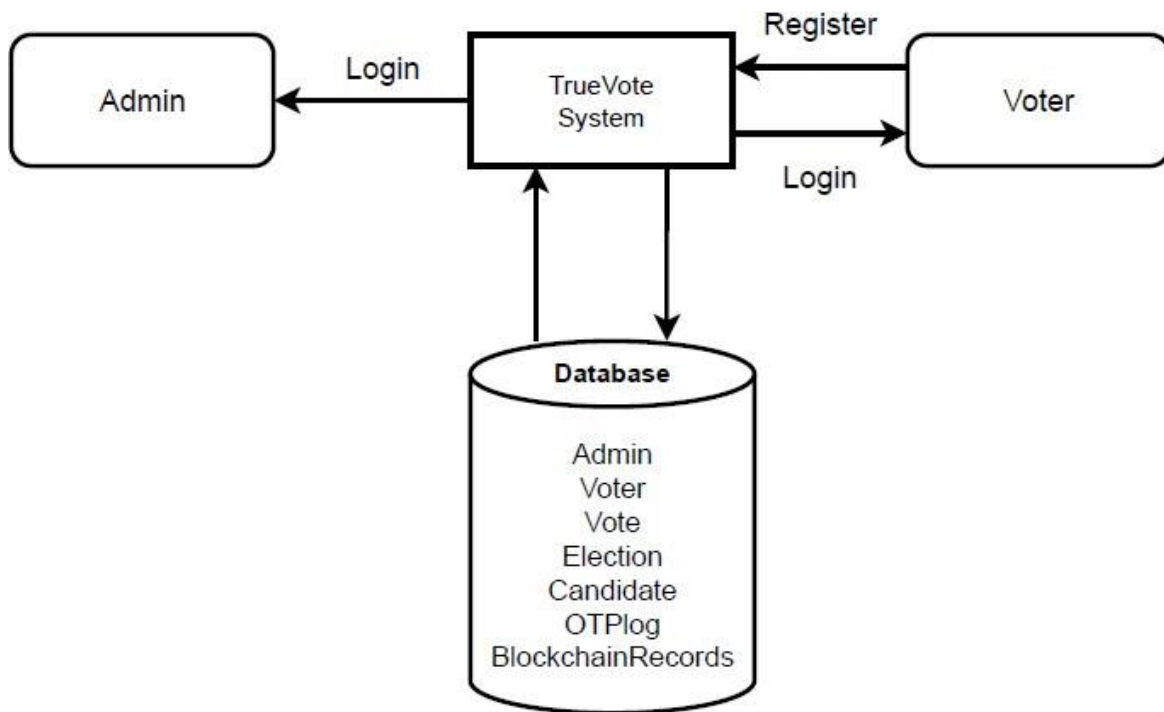


FIGURE 5.1. *System Architecture*

The system architecture of the TrueVote project represents the interaction between key system components — namely the Admin, Voters, the TrueVote Web Application, and the Database integrated with the Blockchain Network.

It defines the logical structure and data flow among these entities to ensure secure, transparent, and tamper-proof election management — from voter registration to result publication.

The architecture adopts a three-tier design — User Interface Layer, Application Layer, and Database Layer with Blockchain Integration — each playing a vital role in maintaining efficiency, scalability, and security.

5.1 User Interface Layer

- This layer provides a user-friendly interface for both Admin and Voters.
- Voters can register, verify OTP, log in, view elections, and cast votes securely through interactive web forms.
- Admins can manage elections, add candidates, verify voters, and publish results.
- The system ensures smooth navigation with role-based access control, preventing unauthorized access to admin or voting features.
- Designed using HTML, CSS, JavaScript, and PHP, the interface ensures responsiveness across mobile and desktop devices.

5.2 Application Layer

- The Application Layer serves as the core processing unit that manages all logical operations and interactions between the UI and the backend database.
- It handles:
 - Authentication & OTP verification
 - Election and candidate management
 - Blockchain transaction initiation for vote recording
 - Vote counting and result retrieval through smart contracts
- When a voter logs in, the system authenticates credentials and ensures that the user is authorized for a specific election.
- Key modules in this layer include:
- Voter Management: Registers and verifies voters.

- Election Management: Admin can create, update, or close elections.
- Blockchain Handler: Connects to the Ethereum smart contract for vote transactions.
- Result Module: Fetches and displays vote counts from blockchain data.
- This layer ensures smooth, real-time communication between the web application, the database and the blockchain network.

5.3 Database Layer

- The Database Layer stores and manages all critical system information using MySQL.
- It includes structured tables such as:
 - Admin – Stores admin credentials and privileges.
 - Voter – Holds voter details, verification status, and login credentials.
 - Election – Contains election metadata and schedule.
 - Candidate – Stores candidate details and linked election IDs.
 - Vote – Records voting activities and status flags.
 - OTPlog – Logs OTP generation and verification attempts.
 - BlockchainRecords – Maintains transaction hashes and timestamps for transparency.
- The database ensures referential integrity, encryption of sensitive data, and high availability through efficient indexing and query optimization.

5.4 Admin Module

- The Admin acts as the supervisory authority of the TrueVote system.
- Responsibilities include:
 - Verifying and approving voter registrations.
 - Creating and managing elections and candidates.

- Monitoring voting progress and blockchain transaction logs.
- Publishing verified results after election closure.
- Admins have exclusive access to analytical reports, system logs, and error monitoring tools to ensure fair and transparent elections.

5.5 Voter Module

- The Voter is the primary user interacting with the TrueVote platform.
- Functionalities include:
- Registration & OTP verification to confirm identity.
- Viewing active elections and candidate lists.
- Casting votes securely — each vote triggers a blockchain transaction recorded on the Ethereum ledger.
- Viewing voting history and blockchain transaction hashes for verification.
- Each voter is restricted to casting one vote per election, enforced through smart contract validation.

5.6 Data Flow and Communication

- The architecture ensures bidirectional communication between the web system, database, and blockchain:
- When a voter registers, details are stored in the database and verified via OTP.
- During voting, the system interacts with the blockchain smart contract to record the vote transaction.
- The transaction hash is then stored in the database for future reference.
- Admins retrieve summarized election data and results directly from both the database and blockchain records.
- This seamless data flow maintains synchronization and transparency across all layers.

5.7 Security and Access Control

- Role-based authentication ensures only authorized users perform sensitive operations.

- Data encryption protects stored voter credentials and transaction details.
- Blockchain immutability prevents tampering or double voting.
- Session management ensures single-login enforcement and automatic logout on inactivity.
- Regular backups and audit trails enhance reliability and accountability.

CHAPTER 6 MODULES

The TrueVote system is built upon a modular architecture designed to ensure secure, transparent, and efficient online voting through blockchain technology. Each module performs a distinct function while maintaining seamless interaction with other components to uphold the integrity, reliability, and usability of the platform. The modular structure enhances scalability and allows independent updates or integration of new features as the system evolves.

6.1 User Management Module

The User Management Module serves as the foundation of the TrueVote system by handling user registration, verification, and profile maintenance. It enables voters, administrators, and election officers to register and manage their credentials securely. During registration, users must provide essential details, including Voter ID, name, and contact information. The module validates this data through database checks and OTP verification to ensure authenticity.

Once verified, users can access personalized dashboards based on their roles. Voters gain access to elections and can cast votes, while administrators can manage election data and monitor activities. This module enforces strict access control, password encryption, and periodic verification to prevent identity fraud. It interacts closely with the Blockchain and Authentication modules to ensure that only legitimate users participate in the voting process.

6.2 Authentication and OTP Verification Module

The Authentication and OTP Verification Module ensures that only authorized and verified users gain access to the system. When a user attempts to log in, their credentials are validated against stored records. To enhance security, a one-time password (OTP) is sent to the registered email or mobile number. This multi-factor authentication ensures that even if credentials are compromised, unauthorized access is prevented.

Once the OTP is verified, the user session is securely created, and role-based access is granted. The module uses encryption algorithms to protect passwords and OTP data in transit and at rest. It also incorporates timeout mechanisms and session expiry to prevent misuse.

This module plays a vital role in maintaining the integrity of the voting process by ensuring that every vote originates from a verified and authenticated voter.

6.3 Election Management Module

The Election Management Module is the administrative backbone of the TrueVote platform. It allows administrators to create, configure, and manage elections within the system. The module enables setting up election titles, dates, eligible voter lists, candidate details, and voting duration. Once an election is created, it becomes visible to authorized voters during the active voting phase.

Administrators can also control the election status—activating, pausing, or closing it when necessary. After the election ends, the module facilitates result compilation and transfers the data to the Blockchain module for verification and permanent recording. This module ensures that elections are transparent, tamper-proof, and efficiently managed throughout their lifecycle.

6.4 Candidate Management Module

The Candidate Management Module focuses on managing details of individuals contesting in elections. It allows administrators to add, update, or remove candidate records, including their names, party affiliations, manifestos, and photos. Candidates' data is linked to specific elections to ensure organized access and display during the voting process.

The module also validates candidate eligibility before approval to prevent duplicate or fraudulent entries. During voting, candidate details are dynamically fetched and presented to voters for selection. After the voting concludes, candidate data is used in the result generation and blockchain verification process. This module's integration with the Election Management and Blockchain modules ensures transparency and consistency in candidate handling.

6.5 Voting and Blockchain Integration Module

The Voting and Blockchain Integration Module is the core of the TrueVote system, ensuring transparency, immutability, and trust in the voting process. When a voter casts a vote, the transaction is immediately encrypted and recorded on the blockchain network. Each vote is stored as a unique transaction hash, making it tamper-proof and verifiable by both

administrators and auditors.

This module ensures that every voter can cast only one vote per election by cross-verifying entries with the database and blockchain ledger. The blockchain component prevents vote duplication and manipulation, ensuring end-to-end integrity of results. The system also provides a transaction hash confirmation to the voter for transparency. This module integrates directly with smart contracts, which automate vote validation, counting, and result generation securely.

6.6 Result and Audit Management Module

The Result and Audit Management Module handles post-election operations, including result computation, verification, and auditing. Once the voting phase ends, the system automatically tallies votes stored on the blockchain using smart contract logic. Since data on the blockchain is immutable, the results are fully verifiable and resistant to tampering.

Administrators can view results through a secure dashboard that displays candidate-wise vote counts and election summaries. The audit function allows third-party verifiers or election commissions to review blockchain records, ensuring complete transparency. The module also provides downloadable reports for archival and legal validation purposes. Through this module, the TrueVote system guarantees that election outcomes are accurate, auditable, and trustworthy.

6.7 Admin Management and Monitoring Module

The Admin Management and Monitoring Module serves as the control center for the entire TrueVote ecosystem. Administrators oversee user verification, election setup, blockchain monitoring, and issue resolution. They have access to analytics dashboards displaying system activity, voter participation rates, and blockchain transaction summaries.

The module includes functionality for suspending fraudulent accounts, resolving disputes, and monitoring system performance in real time. It also enables admins to view voting logs, manage feedback, and generate periodic reports on election activities. This centralized control ensures the smooth operation of the system while maintaining fairness, transparency, and compliance with electoral guidelines.

CHAPTER 7 DIAGRAMS

7.1 Entity Relationship (ER) Diagram

The ER Diagram of the TrueVote system illustrates the relationships between key entities such as Admin, Voter, Election, Candidate, Vote, and OTPLog. Admin manages elections and candidates, while Voters register, verify identity using OTP, and cast votes securely. The Election entity holds details like title, description, and duration, and each election includes multiple candidates. The Vote entity links voters, candidates, and elections, storing information such as transaction hash and timestamp for blockchain verification. The OTPLog entity records OTP verification details to ensure secure authentication. Overall, the diagram shows how all components interact to ensure transparent, verifiable, and tamper-proof digital voting.

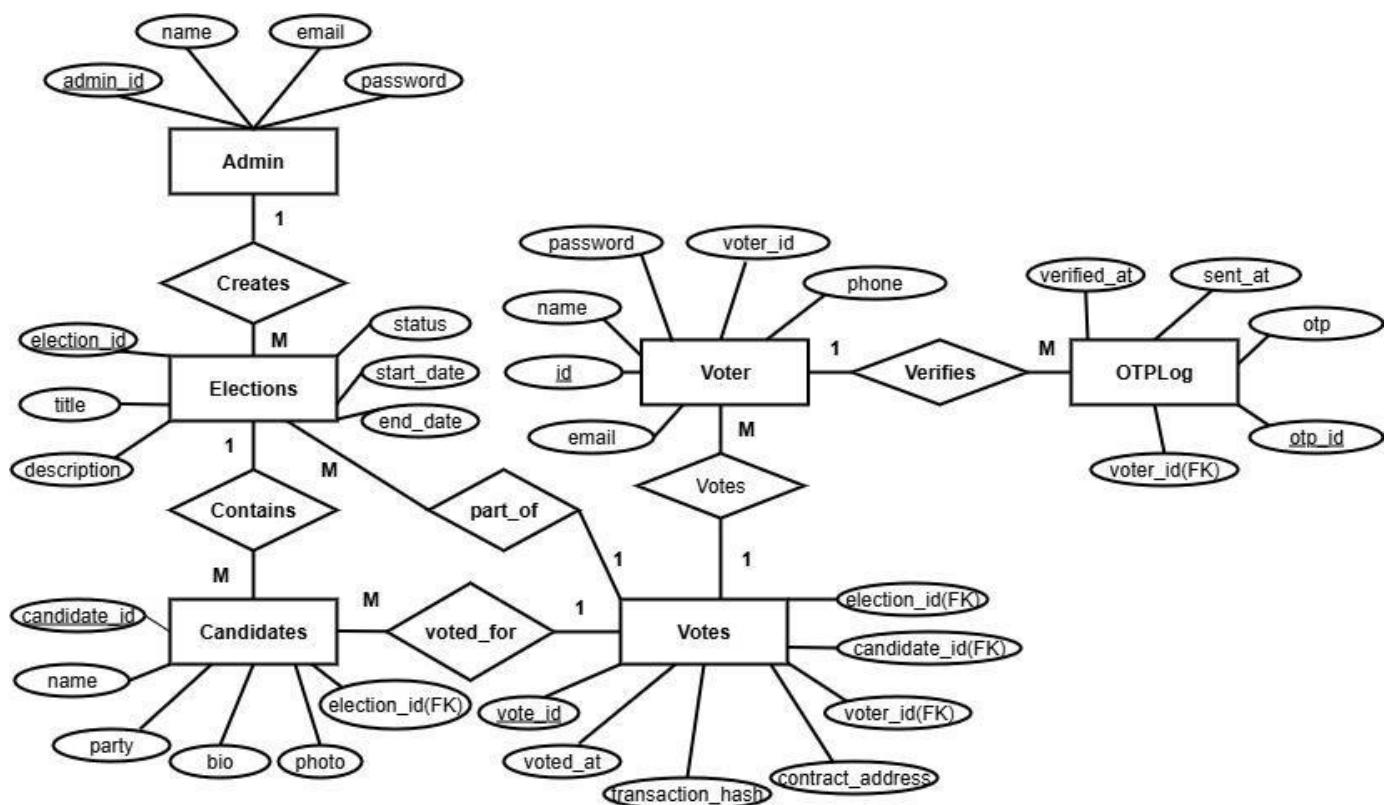


FIGURE 7.1 ER diagram

7.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) of the TrueVote system represents how data flows between different components of the digital voting platform. It illustrates the exchange of information between voters, admins, and the blockchain-based system. The Admin manages elections and candidates, while Voters interact with the system through processes like registration, OTP verification, and casting votes. The system validates inputs, records votes securely on the blockchain, and stores related data such as election details, voter credentials, and transaction logs. Overall, the DFD provides a clear overview of how secure and transparent data processing occurs in TrueVote.

7.2.1 LEVEL 0 DFD

A Level 0 Data Flow Diagram, also known as a context diagram, provides an overall view of the TrueVote system as a single process. It shows how the system interacts with external entities such as Voters and Admins by representing the flow of requests and responses between them. Voters send requests for registration, authentication, and voting, while Admins manage elections and candidate data. The system processes these requests and provides responses accordingly. In simple terms, this diagram defines the boundaries of TrueVote and its relationship with the external environment.

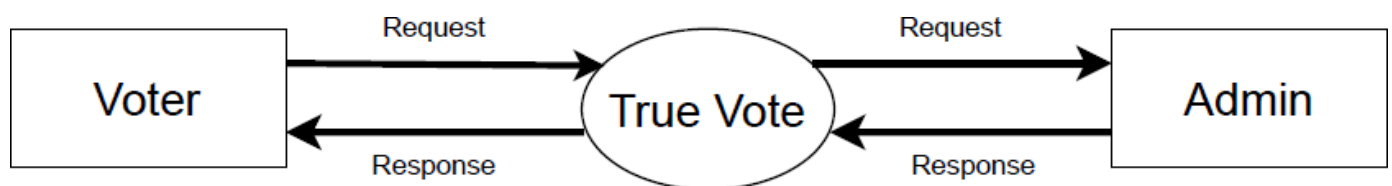


FIGURE 7.2 DFD Level 0

7.2.2 LEVEL 1 DFD VOTER

A Level 1 Data Flow Diagram (DFD) for voter provides a detailed breakdown of the main processes that occur within the system and shows how data flows between the voter, the system's internal components, and the data stores. It expands on the Level 0 DFD by illustrating the internal functions that handle specific voter actions.

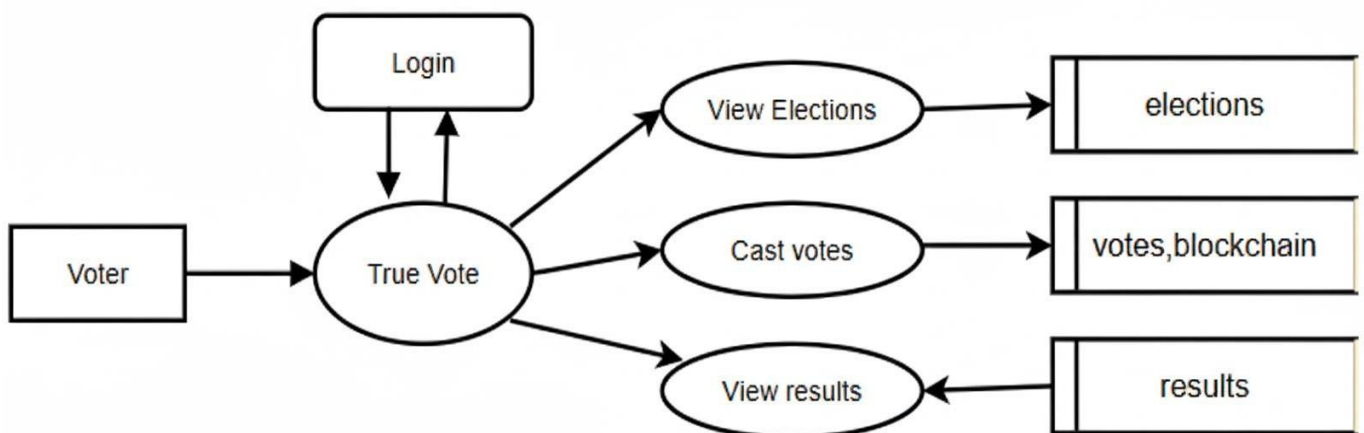


FIGURE 7.3 DFD Level 1 Voter

7.2.3 LEVEL 1 DFD ADMIN

The Level 1 Data Flow Diagram (DFD) for Admin illustrates how the administrator interacts with different internal processes of the system to ensure smooth management, monitoring, and control. It breaks down the main activities the admin performs and shows how data moves between the admin, system modules, and data stores. Thus, the Level 1 DFD for the admin clearly represents how the administrator interacts with multiple processes and databases to maintain order, security, and efficiency across the entire system.

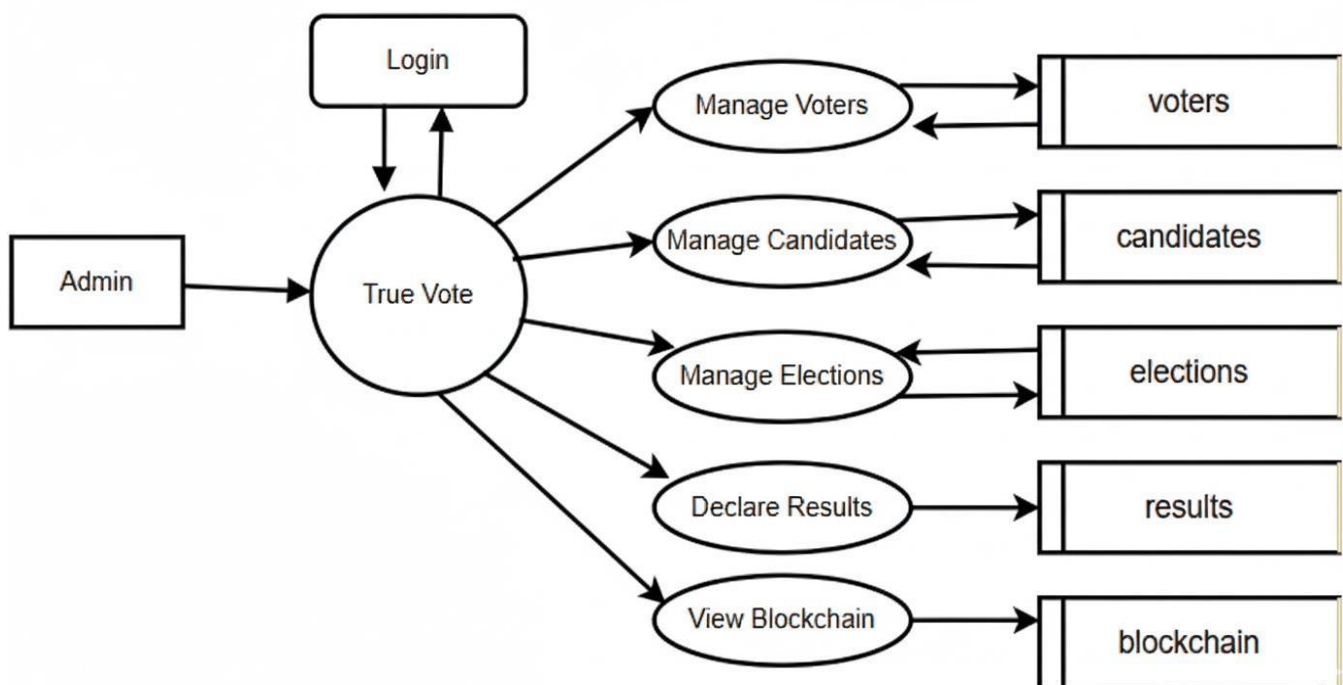


FIGURE 7.4 DFD Level 1 Admin

7.3 USECASE DIAGRAM

A Use Case Diagram visually represents the interactions between users and a system, showing how external actors communicate with the system's main functions. It illustrates the relationships between different roles and the actions they can perform, providing a clear picture of the system's functional scope. The diagram helps to identify who will use the system and what operations they will carry out, making it easier to understand user requirements and system behaviour. Each actor represents a specific type of user or external entity, while each use case describes a goal that the actor wants to achieve through the system.

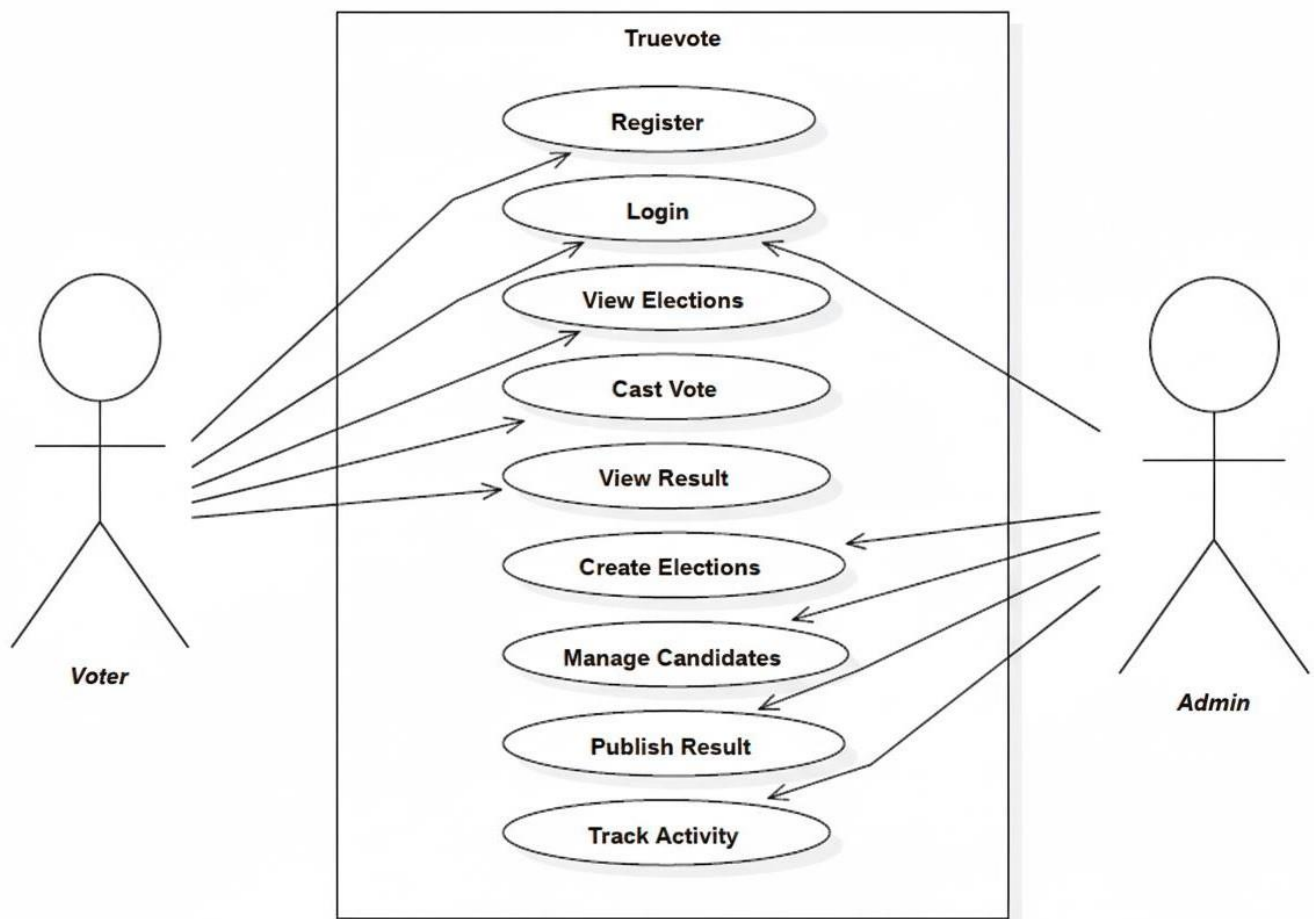


FIGURE 7.5 *Use Case Diagram*

7.4 CLASS DIAGRAM

A Class Diagram provides a detailed structural view of a system by illustrating its classes, attributes, methods, and the relationships between them. It serves as a blueprint for the system's object-oriented design, showing how different entities interact and depend on one another. In a class diagram, each class represents a real-world concept or component within the system, containing its properties and behaviours. The diagram also depicts various relationships such as inheritance, association, aggregation, and composition, which define how classes collaborate to perform system functions.

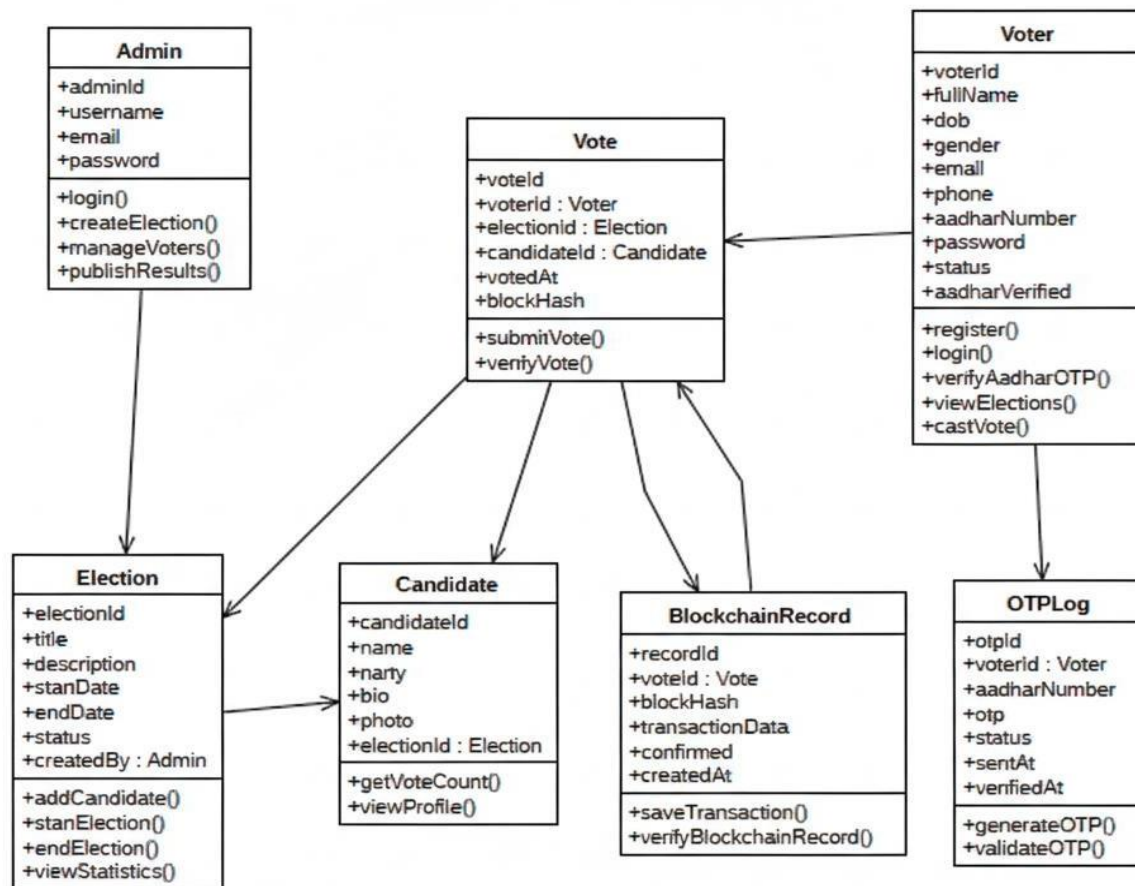


FIGURE 7.6 Class Diagram

7.5 SEQUENCE DIAGRAM

A Sequence Diagram is a type of diagram that illustrates how actors and components interact with each other in a particular sequence over time. It visually represents the flow of messages or interactions between different components or actors in the system to accomplish a specific function or process. In the TrueVote project, the sequence diagram shows how the Admin, Voter, App, Database, and Blockchain components communicate during activities like registration, login, voting, and result publication, highlighting the order and timing of each interaction.

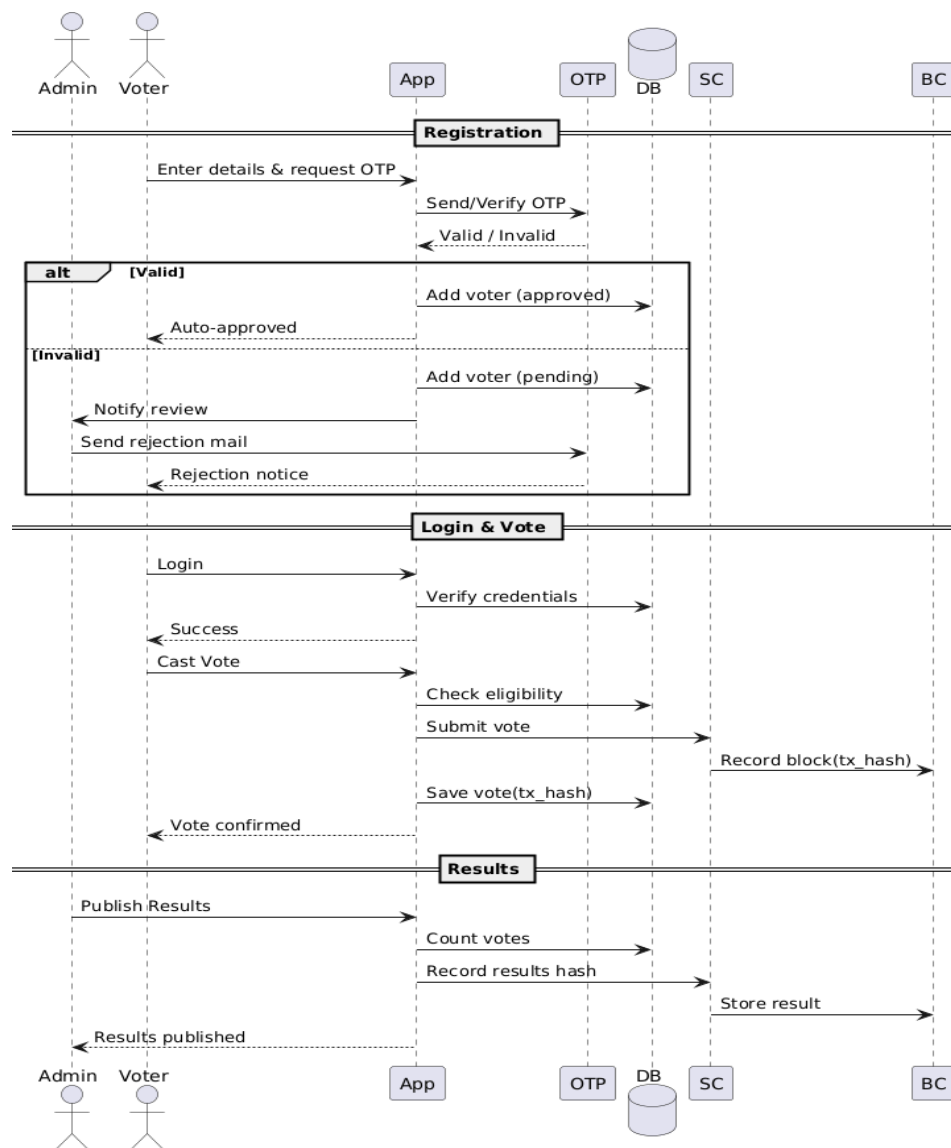


FIGURE 7.7 Sequence Diagram

CHAPTER 8

TESTING

8.1 Data Collection and Preparation

Before testing began, it was essential to collect and prepare relevant datasets and inputs required to simulate real-world voting scenarios for the TrueVote system. The dataset for this project consisted of voter registration details, election information, candidate lists, and OTP verification logs. Each record included attributes like voter ID, election ID, candidate ID, timestamps, and blockchain transaction hashes.

To ensure realistic testing, data representing various user actions were generated — such as multiple voters participating in the same election, repeated OTP verification attempts, and parallel voting transactions. This helped simulate real election load and security scenarios. The data was structured using relational tables such as Voter, Admin, Elections, Candidates, Votes, and OTPLog, ensuring referential integrity and proper linkage between modules.

8.2 Functional Testing

Functional testing was performed to ensure that every function of the TrueVote system operates as per defined requirements. The goal was to validate that all modules — including voter registration, OTP verification, election creation, candidate management, and vote casting — worked correctly and securely.

Functional Testing Steps:

- **User Authentication:** Verified voter registration, login, and OTP verification to ensure only legitimate voters gained access.
- **Election Management:** Tested admin operations such as creating, updating, or closing elections, ensuring election data consistency.
- **Candidate Management:** Checked that admins could add or remove candidates linked to specific elections.

- Voting Process: Ensured each voter could vote only once per election, with every vote recorded immutably on the blockchain.
- Transaction Logging: Confirmed that each successful vote generated a unique transaction hash and contract address stored in the database.

Outcome:

All functions performed successfully under controlled conditions, with minor session management issues fixed during debugging.

8.3 Validation and Performance Testing

Validation and performance testing were crucial to ensure that TrueVote maintained integrity, reliability, and scalability under multiple concurrent voting operations.

8.3.1 Validation Testing:

Validation compared system outputs against specified requirements. The system was validated for OTP verification, single-vote enforcement, and blockchain confirmation.

- Voters were allowed to vote only in active elections linked to their credentials.
- OTP verification accurately authenticated user identity before voting.
- Each blockchain transaction was successfully recorded and retrievable for verification.

8.3.2 Performance Testing:

Performance testing measured system response time and reliability during high user load.

Key areas evaluated included:

- Load Testing: Simulated hundreds of voters voting simultaneously in the same election.
- Response Time: Maintained under 3 seconds for major processes like OTP verification and blockchain transaction confirmation.
- Scalability: The system sustained stable performance up to 500+ concurrent voter sessions.

- Database Stress: Queries for election results and vote verification maintained consistent performance under heavy data load.

Outcome:

The TrueVote system remained stable, responsive, and efficient even during peak usage.

8.4 Integration and User Acceptance Testing

Integration testing validated the smooth flow of data across interconnected modules, ensuring consistency between user, admin, and blockchain components.

8.4.1 Integration Testing:

The following module integrations were tested:

- Voter and OTP Modules: Verified that only OTP-verified voters could access the voting interface.
- Admin and Election Modules: Confirmed that elections created by admins automatically appeared in the voter dashboard.
- Voting and Blockchain: Ensured each vote triggered a blockchain transaction and generated a valid hash.

All integrations executed successfully, maintaining correct data synchronization.

8.4.2 User Acceptance Testing (UAT):

UAT was performed by allowing a group of test voters and admins to use the live prototype. The aim was to evaluate ease of use, transparency, and security perception.

Participants found the interface simple and secure, with clear steps for registration, verification, and voting. Feedback highlighted the need for a clearer success message after blockchain confirmation, which was later implemented.

Result:Users expressed strong confidence in the system’s transparency and usability, confirming readiness for deployment.

8.5 Security and Reliability Testing

Security testing was conducted to ensure data confidentiality, vote integrity, and resistance to manipulation.

8.5.1 Security Measures Tested:

- **Authentication Protection:** Verified password encryption, OTP validation, and prevention of unauthorized dashboard access.
- **Blockchain Security:** Ensured immutability of vote records and verified smart contract integrity.
- **Data Security:** Checked that all sensitive data such as passwords, OTPs, and voter details were securely encrypted.
- **Access Control:** Ensured voters and admins had restricted, role-based access.
- **SQL Injection & XSS:** Tested and confirmed strong resistance to injection and scripting attacks.

8.5.2 Reliability Testing:

Reliability testing verified consistent performance during repeated test cycles. The system maintained a 99% uptime during voting simulations, ensuring accurate vote recording and result generation even under concurrent blockchain operations.

8.6 Deployment and Monitoring

After successful testing, the TrueVote system was deployed on a local and later cloud-based server for pilot execution.

8.6.1 Deployment Steps:

- **Environment Setup:** The final build was deployed using PHP with a MySQL database, integrated with the Ethereum blockchain via smart contracts.

- Data Migration: Dummy test data was replaced with verified voter and election data.
- Version Control: Git was used to track updates and ensure rollback safety.
- Backup and Recovery: Automated daily database backups were configured for security and continuity.

8.6.2 Monitoring and Maintenance:

Post-deployment, the system was continuously monitored for performance, transaction errors, and user issues.

- Error Logs: Automated error tracking facilitated rapid debugging of runtime issues.
- Performance Monitoring: Continuous monitoring ensured stable transaction times and secure blockchain interaction.
- User Feedback Review: Feedback from pilot users helped refine UI elements and enhance verification alerts.

Outcome:

The deployed system operated reliably, with secure, transparent vote recording and user-friendly performance under real-world testing.

CHAPTER 9

ADVANTAGES & DISADVANTAGES

9.1 Advantages

- **Enhanced Security:** TrueVote leverages blockchain technology to ensure that every vote is securely recorded and tamper-proof, eliminating risks of data manipulation or unauthorized access.
- **Transparency and Trust:** Since all voting transactions are stored on a public, immutable ledger, the system promotes complete transparency and builds voter confidence in the election process.
- **Elimination of Electoral Fraud:** The decentralized nature of blockchain prevents vote duplication, fake registrations, and result tampering, ensuring fair and verifiable elections.
- **Efficient Vote Counting:** Votes are automatically recorded and verified through smart contracts, enabling instant and accurate result generation without manual intervention.
- **User-Friendly Participation:** The platform offers a simple and intuitive interface where voters can securely register, verify via OTP, and cast their votes easily using any device with internet access.

9.2 Disadvantages

- **Dependence on Internet Connectivity:** As the system operates online, stable internet access is required. Poor connectivity may delay vote submission or transaction confirmation.
- **Blockchain Transaction Costs:** Deploying and executing smart contracts on blockchain networks may involve gas fees, which could increase operational costs during large-scale elections.
- **Technical Literacy Requirement:** Some voters may find it difficult to understand blockchain-based processes or online voting steps, especially those unfamiliar with digital systems.

- **Scalability Limitations:** Handling a massive number of simultaneous blockchain transactions during national elections could strain system performance and slow down processing times.
- **Data Privacy Concerns:** Although blockchain ensures transparency, storing voter-related data must comply with privacy standards to prevent misuse or unintended exposure of personal information.

CHAPTER 10

RESULT

The results of the TrueVote project demonstrate that the system successfully enables secure, transparent, and verifiable digital voting through blockchain integration. It ensures that every vote is immutably recorded, preventing tampering and duplication while maintaining voter anonymity. The system's OTP-based authentication and smart contract mechanisms guarantee reliable identity verification and automated result generation. With its user-friendly interface, both voters and administrators can easily participate and manage elections. Overall, the project proves effective in providing a trustworthy, efficient, and technologically advanced e-voting solution.

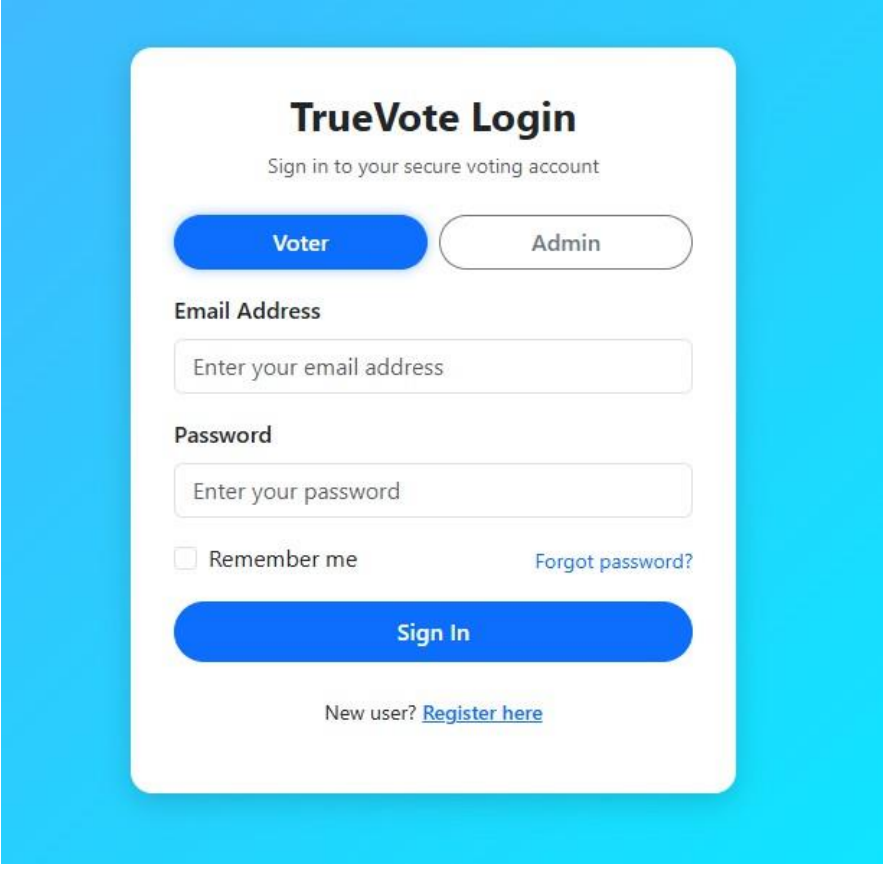
The image shows a login interface for 'TrueVote'. It features a white card with rounded corners on a blue background. At the top, the title 'TrueVote Login' is displayed in bold, followed by the subtitle 'Sign in to your secure voting account'. Below this are two buttons: a blue 'Voter' button and a white 'Admin' button with a blue border. The 'Email Address' section has a text input field with the placeholder 'Enter your email address'. The 'Password' section has a text input field with the placeholder 'Enter your password'. Below the password field is a checkbox labeled 'Remember me' and a link 'Forgot password?'. A large blue 'Sign In' button is positioned below these fields. At the bottom, there is a link 'New user? Register here'.

FIGURE 10.1. *Sign In*

Figure 10.1 is the Sign-In Page for users to access the TrueVote secure online voting system. It allows both Voters and Admins to log in securely using their registered credentials, namely Email Address and Password. This dual-tabbed page ensures authentication by verifying the user's details and role before granting access to personalized dashboards for managing elections or casting votes. The interface is designed to be simple and intuitive, with clear input fields, a "Remember me" option, and links like "Forgot password?" and "Register here" to enhance user convenience and accessibility.

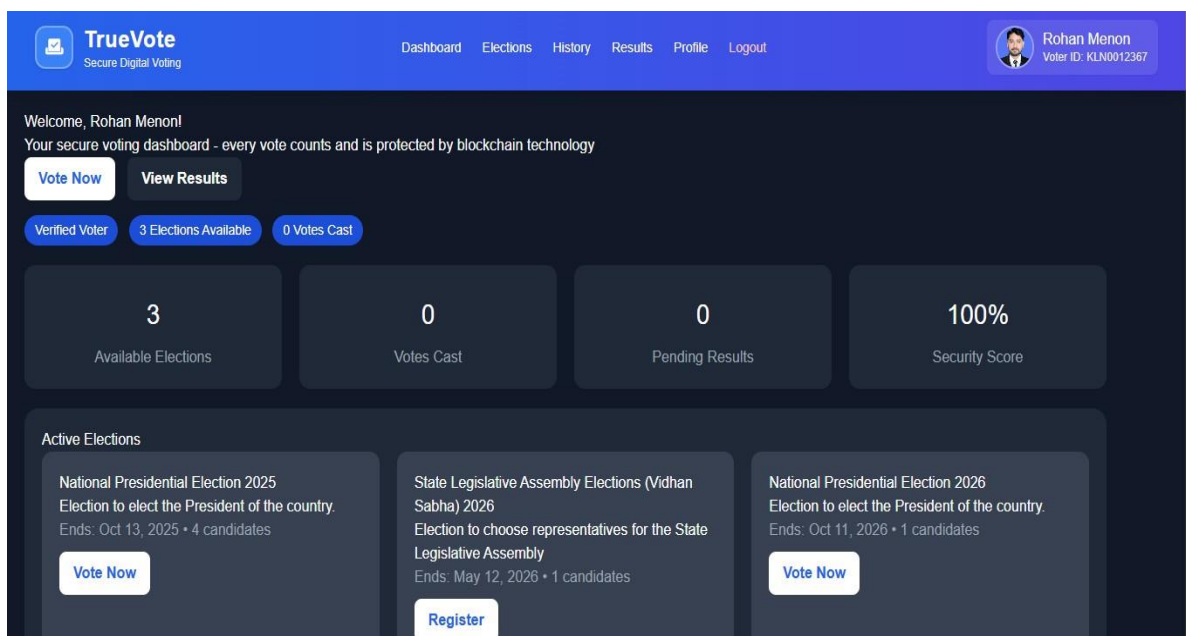


FIGURE 10.2. Voter Page

Figure 10.2 is the Voter Dashboard module, which is a crucial part of the TrueVote system, designed to provide authenticated voters with a personalized and secure interface. It serves as the central hub for the voter, displaying their verified status and key election statistics, including Available Elections, Votes Cast, and Pending Results. Once logged in, voters can access primary functionalities like viewing Active Elections, casting their secure, blockchain-protected vote through the "Vote Now" button, navigating to their History and Results pages, or managing their Profile. This module plays a vital role in maintaining system transparency, protecting the integrity of the voting process, and providing a seamless interface to the platform's core electoral functionalities.

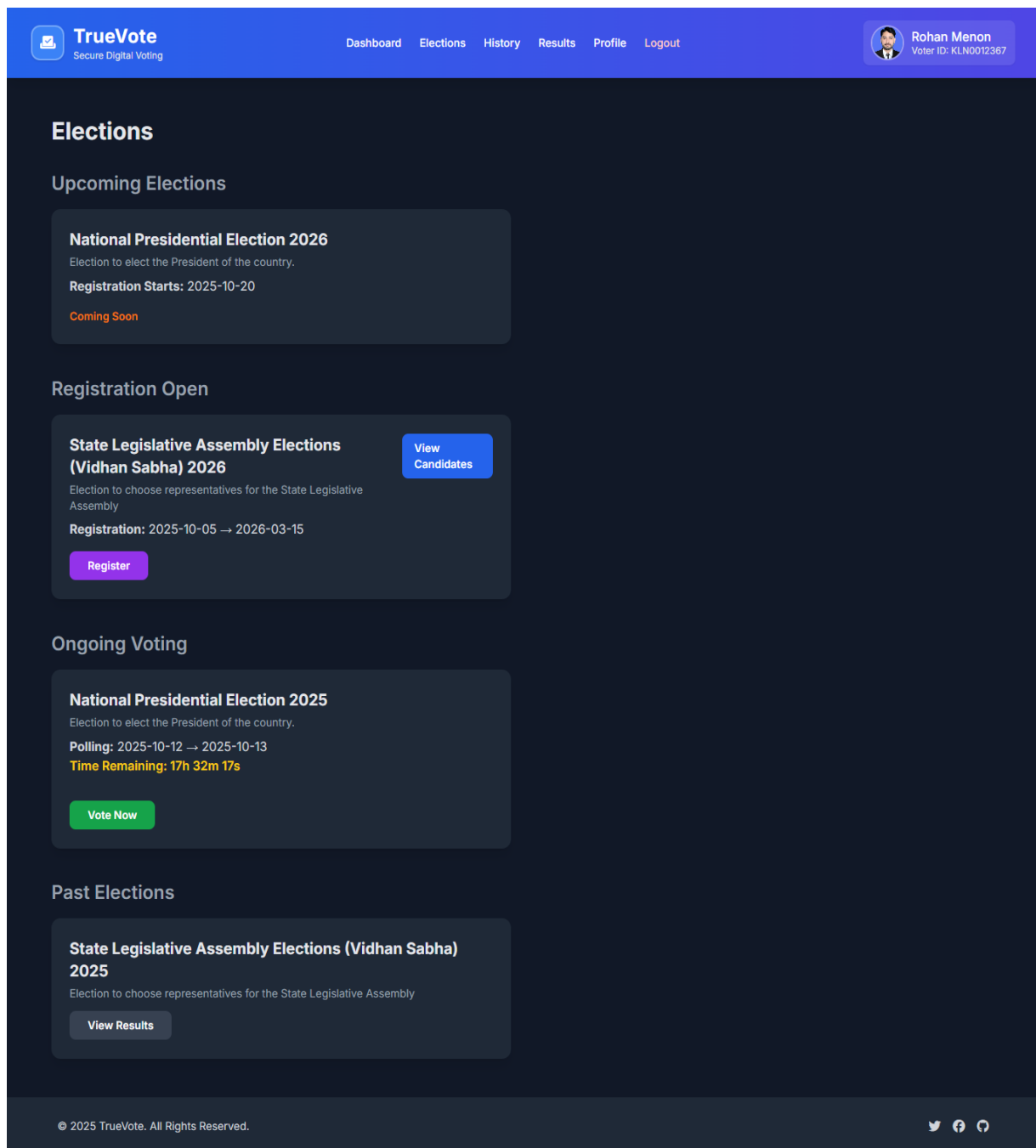


FIGURE 10.3 *Election page*

Figure 10.3 is the Elections Page module, a crucial part of the TrueVote system, designed to provide comprehensive, time-based access to all elections. It ensures that voters can easily navigate and participate by categorizing electoral events into Upcoming, Registration Open, Ongoing, and Past status, thus maintaining transparency and managing the entire electoral lifecycle.

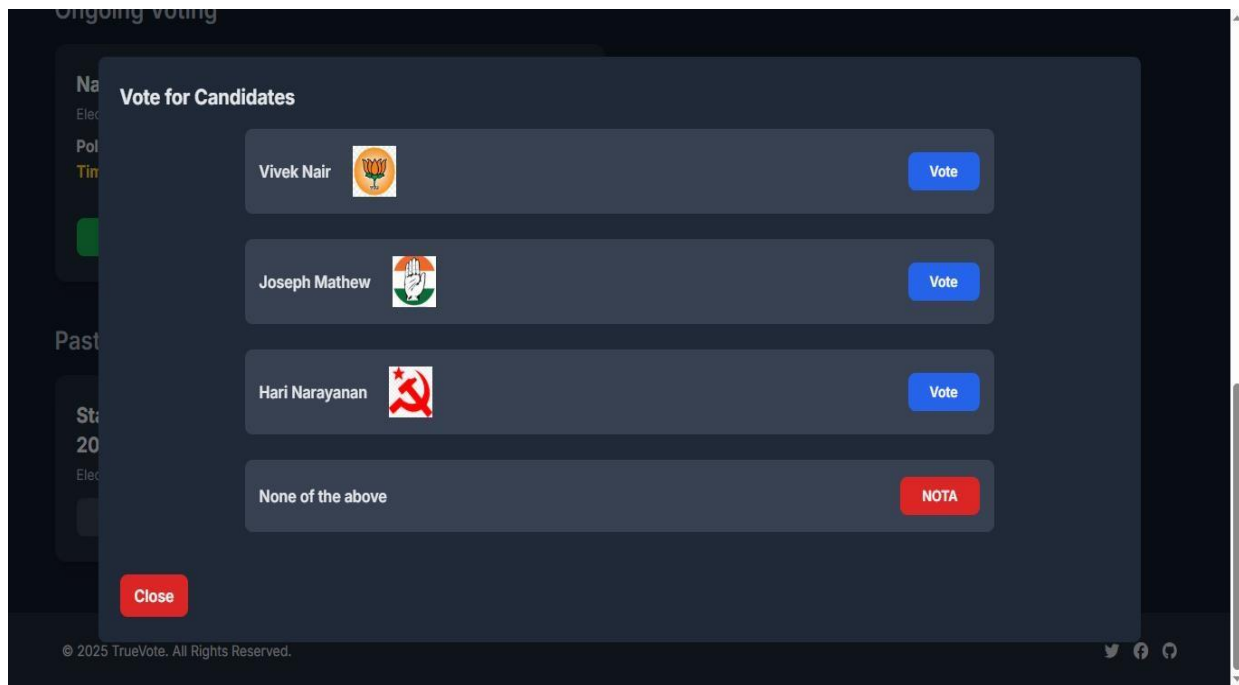


FIGURE 10.4 *Voting page*

Figure 10.4 illustrates the Voting Page of the TrueVote system, where authenticated voters can securely cast their votes. This interface lists the contesting candidates along with their respective symbols and party names, ensuring a familiar and transparent experience similar to traditional ballots. Each candidate is displayed with a dedicated “Vote” button, allowing the voter to make a single, irreversible choice. A “None of the Above (NOTA)” option is also provided to ensure voter freedom and neutrality. Once the vote is submitted, it is instantly recorded as a tamper-proof transaction on the blockchain, guaranteeing transparency, immutability, and verifiability. The design ensures clarity, simplicity, and accessibility, enabling voters to participate effortlessly while maintaining the security and confidentiality of their choices.



FIGURE 10.5 *Result page*

Figure 10.5 illustrates the Result Page of the TrueVote system, which transparently displays the final outcomes of completed elections. Each election result is published automatically after the blockchain verification process confirms vote integrity and immutability. The page presents candidates along with their party symbols, total votes received, and vote percentage, visually represented through progress bars for clarity. The winning candidate is highlighted with a distinct border and “Winner” badge for easy identification. The system also includes the “None of the Above (NOTA)” option, ensuring inclusivity and reflecting all possible voter choices. By leveraging blockchain-backed verification, this module ensures that results are tamper-proof, publicly auditable, and instantly accessible to all stakeholders, enhancing trust and transparency in the electoral process.

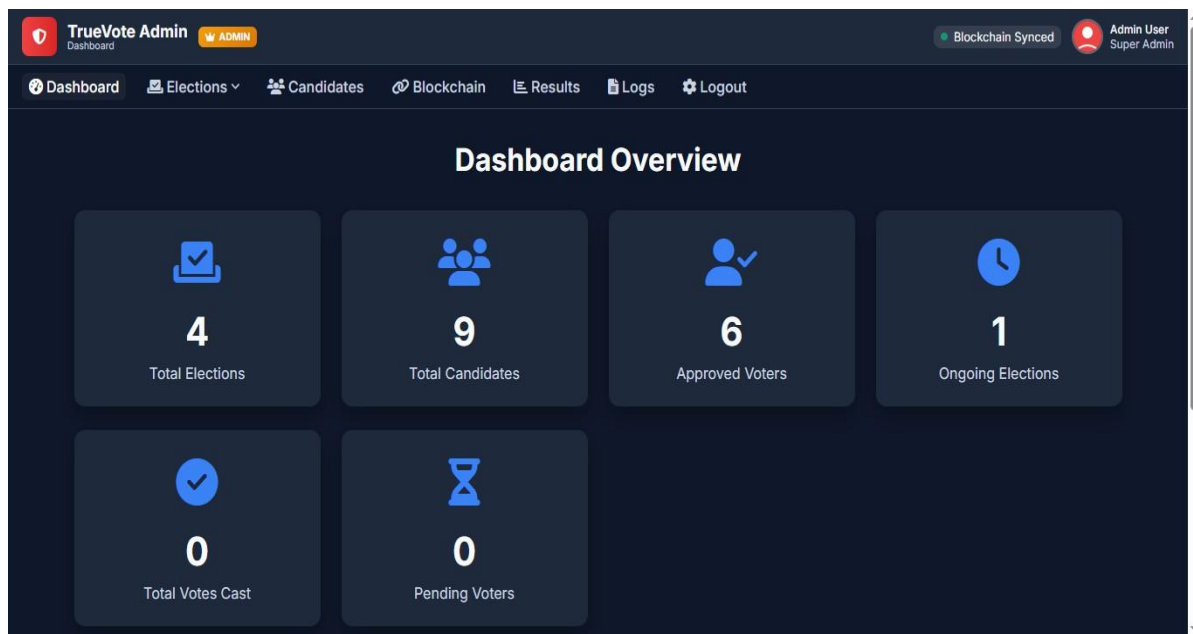


FIGURE 10.6 *Admin Page*

Figure 10.4 is the Admin Dashboard. It serves as the central control panel of the TrueVote system, providing Super Administrators with comprehensive management capabilities. From this interface, the admin can monitor real-time statistics like Total Elections, Approved Voters, and Ongoing Elections, and oversee the integrity of the voting process. The admin can also access modules to manage Candidates, monitor the Blockchain status, declare Results, and review system Logs to ensure smooth and auditable election performance. Additionally, this module enables the admin to maintain data integrity by updating or removing invalid records and ensuring smooth system performance.

CHAPTER 11

CONCLUSION

The TrueVote project represents a major advancement in transforming the traditional voting process through secure, transparent, and decentralized digital technology. By leveraging blockchain and smart contract mechanisms, the system ensures vote immutability, transparency, and tamper-proof recordkeeping, thereby eliminating risks of electoral fraud and unauthorized manipulation. The platform guarantees that each vote is uniquely cast and verifiable while maintaining the confidentiality of voter identity.

Throughout its development, the project demonstrated how blockchain technology can be harnessed to promote trust, efficiency, and accountability in democratic systems. With its intuitive interface, the system simplifies every stage of the election process — from voter registration and OTP verification to vote casting and result publication — ensuring accessibility and reliability for both voters and administrators.

Furthermore, the project emphasizes the growing importance of technological innovation in strengthening democratic participation and digital governance. It highlights how distributed ledger systems can ensure integrity, enhance transparency, and restore public confidence in electoral processes.

In conclusion, the TrueVote project successfully achieves its goal of providing a secure, transparent, and user-friendly e-voting solution. It lays the groundwork for future enhancements such as biometric authentication, real-time result dashboards, and multi-level election integration. Ultimately, the project demonstrates how blockchain can redefine the foundations of trust, transparency, and inclusivity in modern voting systems.

FUTURE SCOPE

The TrueVote system provides a secure and transparent foundation for digital voting, with vast potential for future expansion to enhance scalability, accessibility, and public trust. Upcoming developments may include biometric authentication such as fingerprint or facial recognition to ensure stronger voter identity verification, and a mobile voting application for convenient, on-the-go participation. Integration with government ID databases like Aadhaar or national voter registries can further reduce fake or duplicate registrations.

Future versions could feature advanced analytics dashboards for election authorities to monitor real-time statistics such as voter turnout, region-wise participation, and demographic insights without compromising privacy. The platform can also be extended to manage multiple election types, including general elections, local body polls, and byelections, under a unified blockchain framework.

Security and reliability can be enhanced using decentralized storage (IPFS), AI-based fraud detection, and certified smart contract auditing to ensure system integrity. To promote inclusivity, the interface can incorporate multilingual support and accessibility features such as text-to-speech and adaptive layouts for differently-abled users.

Finally, deploying the system as a blockchain consortium model involving multiple authorized election bodies can promote decentralization, shared governance, and transparent electoral processes—making TrueVote adaptable for national, state, and byelection scenarios with high efficiency and credibility.

APPENDICES

```

voter > contract_deploy.py > ...
1 import sys, json
2 from web3 import Web3
3 from solcx import compile_source, install_solc, set_solc_version
4 from eth_account import Account
5 from config import My_PRIVATE_KEY # Ganache private key
6
7 # Read args
8 if len(sys.argv) != 4:
9     print(json.dumps({"error": "Usage: python contract_deploy.py <election_id> <candidate_id> <user_id>"}))
10    sys.exit(1)
11
12 try:
13     election_id = int(sys.argv[1])
14     candidate_id = int(sys.argv[2])
15     user_id = int(sys.argv[3])
16
17     GANACHE_RPC = "http://127.0.0.1:7545"
18     PRIVATE_KEY = My_PRIVATE_KEY
19
20     election_source = """
21     // SPDX-License-Identifier: MIT
22     pragma solidity ^0.8.17;
23
24     contract IndianElection {
25         uint256 public election_id;
26         uint256 public candidate_id;
27         uint256 public user_id;
28
29         constructor(uint256 _election_id, uint256 _candidate_id, uint256 _user_id) {
30             election_id = _election_id;
31             candidate_id = _candidate_id;
32             user_id = _user_id;
33         }
34     }
35     """
36
37     install_solc("0.8.17")
38     set_solc_version("0.8.17")
39
40     compiled = compile_source(election_source, output_values=["abi", "bin"])

```

Activate Windows
Go to Settings to activate

```

voter > contract_deploy.py > ...
40 compiled = compile_source(election_source, output_values=["abi", "bin"])
41 _, contract_interface = compiled.popitem()
42 abi, bytecode = contract_interface["abi"], contract_interface["bin"]
43
44 w3 = Web3(Web3.HTTPProvider(GANACHE_RPC))
45 assert w3.is_connected(), "❌ Cannot connect to Ganache"
46
47 acct = Account.from_key(PRIVATE_KEY)
48 my_address = acct.address
49 chain_id = w3.eth.chain_id
50 nonce = w3.eth.get_transaction_count(my_address)
51
52 Election = w3.eth.contract(abi=abi, bytecode=bytecode)
53 construct_txn = Election.constructor(election_id, candidate_id, user_id).build_transaction({
54     "from": my_address,
55     "nonce": nonce,
56     "chainId": chain_id,
57     "gas": 2000000,
58     "gasPrice": w3.eth.gas_price
59 })
60
61 signed = acct.sign_transaction(construct_txn)
62 tx_hash = w3.eth.send_raw_transaction(signed.raw_transaction)
63 tx_receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
64
65 result = {
66     "tx_hash": tx_hash.hex(),
67     "contract_address": tx_receipt.contractAddress
68 }
69 print(json.dumps(result)) # 🟢 print only JSON
70
71 except Exception as e:
72     print(json.dumps({"error": str(e)}))
73     sys.exit(1)
74

```

Activate Windows
Go to Settings to activate


```

elections.php IM results.php admin M results.php voter M blockchain.php M X contract_deploy.py
admin > blockchain.php > html > body > script > targetAddress
9 <html lang="en">
50 <body>
53 <div class="container">
54 <h2>Transactions for Ganache Address</h2>
55 <div class="card">
56 <p><strong>Tracking Address:</strong>
57 <span style="color:#3b82f6;">0x2d8868244689b4c6f8ca5d9A05946113FC871ea</span>
58 </p>
59 <button id="loadTx" class="btn-primary"><i class="fas fa-sync-alt"></i> Load Transactions</button>
60 <div id="txList"></div>
61 </div>
62 </div>
63
64 <?php include "includes/footer.php"; ?>
65
66 <script src="https://cdn.jsdelivr.net/npm/web3@1.10.0/dist/web3.min.js"></script>
67 <script>
68 const web3 = new Web3("http://127.0.0.1:7545"); // Ganache RPC
69 const targetAddress = "0x2d8868244689b4c6f8ca5d9A05946113FC871ea".toLowerCase();
70
71 document.getElementById("loadTx").addEventListener("click", async () => {
72 const latestBlockNumber = await web3.eth.getBlockNumber();
73 const txDiv = document.getElementById("txList");
74 txDiv.innerHTML = "<p>Loading transactions...</p>";
75 let found = false;
76 txDiv.innerHTML = "";
77
78 for (let i = 0; i <= latestBlockNumber; i++) {
79 const block = await web3.eth.getBlock(i, true);
80 for (let tx of block.transactions) {
81 if (tx.from.toLowerCase() === targetAddress || (tx.to && tx.to.toLowerCase() === targetAddress)) {
82 found = true;
83 let receipt = await web3.eth.getTransactionReceipt(tx.hash);
84 let toAddress = tx.to ? tx.to : (receipt.contractAddress ? receipt.contractAddress : "N/A");
85
86 const txInfo = `
87 <div class="tx-card">
88 <p><strong>Block:</strong> ${i}</p>
89 <p><strong>Tx Hash:</strong> ${tx.hash}</p>
90 <p><strong>From:</strong> ${tx.from}</p>
91 <p><strong>To / Contract:</strong> ${toAddress}</p>
92 <p><strong>Gas Used:</strong> ${receipt.gasUsed}</p>
93 </div>
94 `;
95 txDiv.innerHTML += txInfo;
96 }
97 }
98 }
99
100 if (!found) {
101 txDiv.innerHTML = "<p>No transactions found for this address.</p>";
102 }
103 });
104 </script>
105 </body>
106 </html>
107

```

Activate Win
Go to Settings to

```

86 const txInfo = `
87 <div class="tx-card">
88 <p><strong>Block:</strong> ${i}</p>
89 <p><strong>Tx Hash:</strong> ${tx.hash}</p>
90 <p><strong>From:</strong> ${tx.from}</p>
91 <p><strong>To / Contract:</strong> ${toAddress}</p>
92 <p><strong>Gas Used:</strong> ${receipt.gasUsed}</p>
93 </div>
94 `;
95 txDiv.innerHTML += txInfo;
96 }
97 }
98 }
99
100 if (!found) {
101 txDiv.innerHTML = "<p>No transactions found for this address.</p>";
102 }
103 });
104 </script>
105 </body>
106 </html>
107

```

```

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187

```

```

<?php
// Fetch candidates for this election
$cand_query = "SELECT * FROM candidates WHERE election_id = ?";
$cand_stmt = $pdo->prepare($cand_query);
$cand_stmt->execute([$election['id']]);
$candidates = $cand_stmt->fetchAll(PDO::FETCH_ASSOC);

// Count votes
$total_votes = 0;
$candidate_votes = [];
foreach ($candidates as $candidate) {
    $vote_query = "SELECT COUNT(*) as votes
                  FROM votes
                  WHERE candidate_id = ? AND election_id = ? AND vote_count = '1'";
    $vote_stmt = $pdo->prepare($vote_query);
    $vote_stmt->execute([$candidate['id'], $election['id']]);
    $votes = $vote_stmt->fetch()['votes'];
    $candidate_votes[$candidate['id']] = $votes;
    $total_votes += $votes;
}

// Prepare candidates with percentages
$candidates_with_votes = [];
foreach ($candidates as $candidate) {
    $votes = $candidate_votes[$candidate['id']];
    $percentage = $total_votes > 0 ? round(($votes / $total_votes) * 100, 2) : 0;
    $candidate['votes'] = $votes;
    $candidate['percentage'] = $percentage;
    $candidates_with_votes[] = $candidate;
}

// Sort by percentage descending
usort($candidates_with_votes, function($a, $b) {
    return $b['percentage'] <=> $a['percentage'];
});

```

Activate Wi
Go to Settings

```

183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

```

```

// Sort by percentage descending
usort($candidates_with_votes, function($a, $b) {
    return $b['percentage'] <=> $a['percentage'];
});

// Determine the winner (highest percentage, excluding NOTA if there are other candidates)
$winner = null;
$max_percentage = 0;
foreach ($candidates_with_votes as $candidate) {
    if ($candidate['percentage'] > $max_percentage && strtolower($candidate['name']) != 'none of the above') {
        $max_percentage = $candidate['percentage'];
        $winner = $candidate;
    }
}

// If all candidates are NOTA or tie, check if NOTA has votes
if (!$winner && !empty($candidates_with_votes)) {
    $winner = $candidates_with_votes[0]; // First in sorted list (highest percentage)
}
?>

```

REFERENCES

- **Ali, R., Anwar, Z., & Khan, S.** (2020): *Decentralized E-Voting System Using Ethereum Blockchain*. [Source: IEEE Xplore]
- **Kshetri, N., & Voas, J.** (2020): *Blockchain-Enabled E-Voting*. [Source: IEEE Computer]
- **Hardwick, F. S., Akram, R. N., & Markantonakis, K.** (2020): *Fair and Transparent Blockchain E-Voting System*. [Source: IEEE Transactions on Emerging Topics in Computing]
- **Patel, V., & Shah, K.** (2021): *Institutional E-Voting Framework Using Private Blockchain*. [Source: International Journal of Advanced Computer Science and Applications (IJACSA)]
- **Noor, T. H., Zeadally, S., & Alfazi, A.** (2023): *Lightweight Blockchain for Secure and Scalable Voting Systems*. [Source: Future Generation Computer Systems, Elsevier]
- **Russo, A., Fernández Anta, A., González Vasco, M. I., & Romano, S. P.** (2021): *Chirotonia: A Scalable and Secure E-Voting Framework Based on Blockchains and Linkable Ring Signatures*. [Source: arXiv preprint]
- **Onur, C., & Yurdakul, A.** (2022): *ElectAnon: A Blockchain-Based, Anonymous, Robust, and Scalable Ranked-Choice Voting Protocol*. [Source: arXiv preprint]
- **Stančíková, I., & Homoliak, I.** (2022): *SBvote: Scalable Self-Tallying Blockchain-Based Voting*. [Source: arXiv preprint]
- **Kiashemshaki, K., Chukwuani, E. N., Jalili Torkamani, M., & Mahmoudi, N.** (2025): *Secure and Scalable Blockchain Voting: A Comparative Framework and the Role of Large Language Models*. [Source: arXiv preprint]
- **Anonymous.** (2025): *Leveraging Blockchain for Robust and Transparent E-Voting Systems*. [Source: ScienceDirect – Journal Article]
- **Anonymous.** (2024): *Blockchain-Based E-Voting Systems: A Systematic Literature Review*. [Source: SSRN – Peer-Reviewed Survey]

- **Anonymous.** (2024): *Blockchain for Securing Electronic Voting Systems: A Survey of Architectures, Trends, Solutions, and Challenges*. [Source: SpringerLink – Journal Article]
- **Anonymous.** (2024): *E-Voting System Using Cloud-Based Hybrid Blockchain Technology*. [Source: ScienceDirect – Journal Article]
- **Anonymous.** (2024): *Blockchain-Based Electronic Voting Systems: A Case Study in Morocco*. [Source: ScienceDirect – Journal Article]
- **Anonymous.** (2024): *Enhancing Security and Transparency in Online Voting Through Blockchain Decentralization*. [Source: Springer – Journal Article]
- **Alzahrani, A., Bulusu, N., & Shiva, S.** (2021): *A Secure Electronic Voting System Using Blockchain Technology*. [Source: IEEE Access]
- **Praveen Kumar, P. S., et al.** (2022): *Blockchain-Based E-Voting: A Review of Technologies and Implementation Models*. [Source: Elsevier]
- **Farag, A., Mousa, M., & El-Bakry, H.** (2023): *Secure Online Voting Using Blockchain with Biometric Integration*.
- **Hossain, M. S., et al.** (2024): *Trustworthy E-Voting Using Blockchain: Challenges and Future Scope*. [Source: Springer]
- **Singh, A., & Kumar, S.** (2025): *Lightweight Blockchain E-Voting System for Educational Institutions*. [Source: IJCSNS]

AMAL SHAJI

Mini project Report

 MINIPROJECT REPORT PLAGARISM CHECKING





Document Details

Submission ID**trn:oid:::10159:117261269****Submission Date****Oct 18, 2025, 9:24 AM GMT+5:30****Download Date****Oct 20, 2025, 11:33 AM GMT+5:30****File Name****Mini project Report (1) (4).pdf****File Size****1.4 MB****62 Pages****11,259 Words****72,478 Characters**




19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **200** Not Cited or Quoted 19%
Matches with neither in-text citation nor quotation marks
-  **0** Missing Quotations 0%
Matches that are still very similar to source material
-  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 6%  Internet sources
- 6%  Publications
- 18%  Submitted works (Student Papers)