Transparent Voting System

A PROJECT REPORT
for
Project (KCA451)
Session (2024-25)

Submitted by

Ritesh Kumar
2300290140143
Satyam Baranwal
2300290140162
Pulkit Jain
2300290140130
Ritik Vihan
2300290140144

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Neha Tyagi
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(APRIL 2025)

CERTIFICATE

Certified that **Ritesh Kumar Roll No. 2300290140143, Satyam Baranwal Roll No. 2300290140162, Pulkit Jain Roll No. 2300290140130, Ritik Vihan Roll No. 230029014014** have carried out the project work having "**Transparent Voting System** " (**Project-KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Ritesh Kumar (2300290140143)
Satyam Baranwal (2300290140162)
Pulkit Jain (2300290140130)
Ritik Vihan (2300290140144)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ms. Neha Tyagi	**Dr. Akash Rajak**
Associate Professor	**Dean**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

Transparent Voting System
ABSTRACT

The Transparent Voting System is a revolutionary initiative aimed at transforming the conventional voting process through the power of blockchain technology, bringing unparalleled transparency, security, and accessibility to democratic elections. In the traditional electoral process, challenges such as electoral fraud, vote tampering, low voter turnout, delayed result processing, and lack of verifiable transparency have often undermined public trust. This system is specifically designed to eliminate these persistent issues by offering a highly secure, decentralized, and immutable platform that allows eligible citizens to cast their votes from anywhere, without the need to visit physical polling stations.

At the core of the Transparent Voting System is a blockchain-based ledger that ensures each vote and every piece of voter data is encrypted, securely stored, and permanently recorded. Due to blockchain's decentralized nature, no single authority can alter or manipulate the data, thereby making the entire election process auditable and tamper-proof. This immutable ledger guarantees that once information is added to the chain, it cannot be changed or deleted, thus safeguarding the integrity of the election.

The registration phase, which is one of the most crucial aspects of the system, is carefully managed by authorized administrators through a controlled interface. These officials are responsible for collecting valid documentation such as government-issued identity proofs, verifying eligibility criteria like age and citizenship, and ensuring that only authentic users are onboarded into the system. Upon successful verification, each user is assigned a unique, encrypted digital Voter ID that serves as a secure identity credential, enabling the individual to log in, participate in elections, and cast their vote through the system.

This digital Voter ID is tightly bound to the user's verified identity, preventing duplication, impersonation, or misuse. Furthermore, if any irregularities, false documents, or unauthorized attempts are detected during the registration process, the system instantly flags and rejects the attempt, safeguarding the voter database against corruption and maintaining the purity of the electoral roll.

Once registered, the voting procedure becomes extremely streamlined—voters can log into the system using their credentials and cast their votes through a simple, intuitive interface. As soon as a vote is submitted, it is instantaneously recorded on the blockchain, creating a transparent, traceable transaction that is permanently visible in the public ledger without disclosing the identity of the voter, thus maintaining voter anonymity while ensuring auditability.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms**. **Neha Tyagi** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Akash Rajak**, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends who have helped me a lot in many critical situations.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Ritesh Kumar

Satyam Baranwal

Pulkit Jain

Ritik Vihan

TABLE OF CONTENTS

	Certificate	i
	Abstract	ii
	Acknowledgements	iii
	Table of Contents	iv-vii
	List of Tables	vii
	List of Figures	viii
1	**Introduction**	1-14
	1.1 Overview	1
	1.1.1 Purpose of the System	2
	1.1.1.1 Need for Transparency in Voting	2
	1.1.1.2 Role of Blockchain Technology	3
	1.1.2 Scope of the Project	4
	1.1.2.1 Target Users	5
	1.2 Problem Statement	7
	1.2.1 Limitations of Traditional Voting	7
	1.2.2 Challenges in Digital Adoption	8
	1.3 Objectives of the Project	10
	1.4 Methodology	11
	1.5 Tools & Technologies Used	12
2	**Feasibility Study**	15-18
	2.1 Technical Feasibility	15

	2.2	Economic Feasibility	15
	2.3	Legal Feasibility	16
	2.4	Operational Feasibility	16
	2.5	Schedule Feasibility	17
3	**System Requirements and Specifications**		**19-21**
	3.1	System requirement specification	19
	3.2	Specific requirement	19
	3.3	Hardware specification	19
	3.4	Software specification	20
	3.5	Functional requirement	20
	3.6	Non-functional requirement	21
	3.7	Performance requirement	21
4	**System Analysis**		22-23
	4.1	Existing System	22
	4.2	Proposed System	22
5	**System Design**		24-29
	5.1	Project modules	24
	5.2	Use-Case diagram	24
	5.3	Dataflow diagram	26
		5.3.1 Level 1 dataflow diagram	27
		5.3.2 Level-2 dataflow diagram	28
6	**IMPLEMETATION**		31-33
	6.1	Concept	31
	6.2	Algorithms	31
	6.3	Functional Modules	32
7	**Code**		33-45

8	**Result**	46-48
9	**TESTING**	49-51
	9.1 Methods of Testing	49
	9.1.1 Unit Testing	49
	9.1.2 Integration Testing	49
	9.1.3 Validation Testing	49
	9.1.4 User Acceptance Testing	50
	9.2 **Test cases**	50
10	**PERFORMANCE ANALYSIS**	52-53
11	**CONCLUSION**	54
12	**REFERENCE**	55

LIST OF TABLES

Table No.	Name of Table	Page
7.1	Function Name and Test results	32
7.2	Input Aadhar Test Case	34
7.3	Email OTP authentication test case	81
7.4	Private key verification Test case	83

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
5.1	System architecture	5
5.2	Case Diagram	5
5.3	Flow chart	6

5.4	Level 1 data flow	7
5.5	Level 2 data flow	8
6.2	Model hash working	11
6.4	OTP validation	40
6.5	Private key authentication	45
6.7	Casting vote	55
8.1	Performance Analysis	57
A.1	Home page	58
A.2	Add voter	59
A.3	Add political party	60
A.4	PHP Admin	62
A.5	Change admin password	63
A.6	To modify voter details	63
A.7	To modify the political party details	87

CHAPTER 1

INTRODUCTION

1.1 Overview

The Transparent Voting System is an advanced digital solution designed to revolutionize the traditional voting process by utilizing the core features of blockchain technology—decentralization, immutability, and transparency. In conventional voting systems, several challenges persist, such as voter fraud, manipulation of results, lack of transparency, delayed counting, and limited accessibility. These issues can raise questions about the integrity of elections and reduce public trust in democratic processes. Blockchain offers a robust and efficient alternative to address these limitations and ensure a secure and verifiable voting environment.

At its core, blockchain is a distributed ledger technology that stores data across multiple nodes in a network. In the context of voting, each vote is treated as a transaction and is recorded on a block. Once validated, the block is added to a chain of previous blocks, forming a permanent and unchangeable record. This eliminates the risk of tampering or deletion of votes after they have been cast. Moreover, all stakeholders—including election commissions, political parties, and voters—can monitor the process in real-time, enhancing transparency.

The system incorporates cryptographic techniques to ensure that voter identities remain anonymous while still enabling vote verification. This dual feature of privacy and accountability is one of the key strengths of blockchain in voting systems. Voters can also receive confirmation that their vote has been recorded correctly without revealing their identity or choice.

Smart contracts can be integrated to automate various phases of the election, such as voter registration, eligibility verification, and result calculation. This reduces the need for manual oversight and lowers the chances of human error or bias.

Additionally, the system can be made accessible via web or mobile applications, making it more convenient for people in remote areas or with limited mobility to participate in elections. This has the potential to increase voter turnout and strengthen democratic engagement.

In conclusion, the Transparent Voting System represents a future-ready approach to electoral integrity. By ensuring secure, transparent, and immutable voting records, it not only safeguards democratic values but also builds public confidence in the electoral process.

1.1.1 Purpose of the System

The primary purpose of the **Transparent Voting System** is to create a secure, transparent, efficient, and tamper-proof digital voting platform that enhances the integrity of the electoral process. With growing concerns over electoral fraud, vote manipulation, delayed results, and lack of trust in traditional voting methods, there is a pressing need for a system that can address these issues effectively. This project aims to leverage the decentralized and immutable nature of blockchain technology to ensure that every vote is accurately recorded and cannot be altered or deleted after submission.

The system is designed to instill confidence among voters by making the entire voting process transparent and verifiable. Each vote is treated as a unique transaction on the blockchain network, and once it is verified, it becomes a permanent part of the blockchain ledger. This ensures that no authority, hacker, or unauthorized person can modify the voting data after the fact, eliminating the risk of tampering and fraud.

Another key purpose of the system is to increase accessibility and voter participation. The platform can be accessed from remote locations through secure web or mobile applications, which allows citizens to vote from anywhere without the need to travel to polling stations. This is especially useful for elderly citizens, people with disabilities, and those residing abroad.

Moreover, the system maintains voter anonymity through cryptographic techniques, ensuring that votes are confidential while still allowing for individual vote verification. This dual approach supports democratic principles by protecting the privacy of the voter while maintaining the transparency of the process.

Automation is another critical purpose of this system. By using smart contracts, processes such as voter registration, eligibility checks, vote counting, and result declaration can be carried out without manual intervention, reducing human error and administrative burden.

Ultimately, the Transparent Voting System aims to modernize the voting infrastructure, making it more reliable, secure, inclusive, and trustworthy. It is not just a technological solution but a step toward strengthening democracy by promoting free and fair elections. By fulfilling these purposes, the system serves as a robust foundation for future digital governance and citizen empowerment in the democratic process.

1.1.1.1 Need for Transparency in Voting

Transparency in voting is one of the most crucial elements of any democratic process. It ensures that every citizen’s vote is counted accurately and that the outcome truly reflects the will of the people. A lack of transparency can lead to mistrust, decreased voter turnout, electoral fraud, manipulation of results, and ultimately, a weakening of democratic institutions. Therefore, ensuring transparency is not just a matter of process but a foundation of public trust and democratic legitimacy.

In many traditional voting systems, especially those relying on paper ballots or centralized electronic machines, transparency is often limited. Voters have little to no insight into what happens after they cast their vote. They cannot verify whether their vote was recorded correctly, counted accurately, or stored securely. This opaque process can lead to suspicion and allegations of vote tampering, especially in closely contested elections.

Furthermore, centralization of voting data and results creates a single point of failure. If compromised—whether due to technical glitches, insider manipulation, or cyberattacks—the entire election's credibility can be questioned. In developing nations or regions with a history of electoral corruption, this lack of transparency can lead to civil unrest, protests, or even violence.

Introducing transparency into the voting process helps prevent such issues. When voters can verify that their vote has been counted without compromising their privacy, it instills confidence in the system. Transparent systems also allow independent audits and real-time monitoring, ensuring that no unauthorized changes are made during or after voting.

Blockchain technology offers a powerful solution to enhance transparency in voting. By recording each vote on a distributed and immutable ledger, it ensures that once a vote is cast, it cannot be changed or deleted. The decentralized nature of blockchain prevents any single entity from controlling or manipulating the outcome, and the ledger can be made publicly viewable (with privacy preserved), allowing anyone to audit the results.

In conclusion, transparency is essential to uphold the principles of fairness, accountability, and trust in elections. As societies move toward digital governance, adopting transparent technologies like blockchain for voting is not just an innovation—it's a necessity. It ensures that democracy evolves with technology while preserving its core values.

1.1.1.2 Role of Blockchain Technology

Blockchain technology plays a transformative role in modernizing the voting process by addressing key challenges such as security, transparency, trust, and data integrity. As a decentralized and distributed digital ledger, blockchain ensures that all voting-related data is stored in a secure, tamper-proof, and transparent manner. This makes it an ideal technology for creating a reliable and trustworthy voting system that enhances the democratic process.

One of the most important contributions of blockchain is **immutability**. Once a vote is recorded on the blockchain, it cannot be altered, deleted, or tampered with. Each vote is stored as a unique transaction within a block, and these blocks are linked cryptographically in chronological order. This chain of records is distributed across multiple nodes (computers) in the network, ensuring that no single authority or hacker can manipulate the data without being detected.

Another crucial aspect of blockchain is **decentralization**. Traditional voting systems rely on centralized databases controlled by election authorities, which makes them vulnerable to internal misuse or external cyberattacks. In contrast, blockchain distributes control across a peer-to-peer network, reducing the risk of a single point of failure and making the system more robust and secure.

Transparency is also a key benefit. Blockchain allows all stakeholders—voters, officials, and observers—to monitor the election process in real-time. While maintaining voter anonymity through cryptographic techniques, the system ensures that every vote is publicly verifiable. Voters can independently confirm that their vote has been counted, increasing trust and participation in the electoral process.

Furthermore, **smart contracts** can be integrated to automate processes such as voter authentication, vote tallying, and result announcement. These contracts are self-executing codes on the blockchain that ensure rules are followed without the need for human intervention, reducing the chances of human error or bias.

Blockchain also enhances **accessibility** by enabling secure remote voting through authenticated devices, allowing citizens to participate even from distant or rural areas.

In summary, blockchain technology provides the foundation for a secure, transparent, and trustworthy voting system. Its role is pivotal in addressing long-standing issues in electoral systems and paving the way for digital democracy. By ensuring data integrity, enhancing trust, and enabling verifiability, blockchain serves as the backbone of a next-generation voting platform.

1.1.2 Scope of the Project

The scope of the **Transparent Voting System** encompasses the design, development, and deployment of a secure and decentralized digital voting platform that addresses the limitations of traditional voting systems. This project aims to leverage blockchain technology to ensure a transparent, tamper-proof, and verifiable election process that is accessible to all eligible voters.

At its core, the project focuses on developing a web-based or mobile application that allows users to cast their votes securely from any location. The system will use blockchain to record each vote as a unique, immutable transaction, thereby preventing duplication, alteration, or deletion of votes. This ensures that every vote is counted exactly once and cannot be manipulated at any stage of the voting process.

The project will implement strong authentication mechanisms to verify the identity of voters while preserving voter anonymity. This includes the integration of cryptographic techniques and user verification methods such as One-Time Passwords (OTP), biometric authentication, or digital signatures. Only verified users will be able to participate in the election, thereby eliminating the chances of fake or duplicate voting.

The system will also include an administrative interface for election officials to manage the election process. This includes configuring election parameters, registering candidates, monitoring the live voting process, and declaring results. Smart contracts will be utilized to automate key tasks such as vote counting and result generation, reducing manual errors and increasing overall efficiency.

The scope also covers real-time monitoring and public auditability. Stakeholders, including voters, political parties, and watchdog organizations, can monitor the voting process transparently, which increases public trust in the system.

Additionally, the system will be designed to handle different types of elections—such as student body elections, organizational voting, municipal polls, and even national elections—with appropriate scalability and security measures.

However, the scope does not include offline or paper-based voting integration, as the focus is solely on digital transformation through blockchain. The project also assumes that the target users have access to the internet and basic digital literacy.

In conclusion, this project’s scope covers the creation of a fully functional, secure, and scalable online voting platform that uses blockchain to bring transparency, security, and efficiency to the democratic process, making it suitable for adoption by institutions and governments seeking modern electoral reforms.

1.1.2.1 Target Users

The **Transparent Voting System** is designed to serve a diverse range of users who are involved at various levels of the voting process. The main aim is to provide a digital, secure, and trustworthy voting environment that can be adopted in multiple contexts—be it national elections, institutional voting, corporate decision-making, or community-level polls. The system’s features are built to support inclusivity, transparency, and ease of use for all stakeholders.

1. **General Voters / Citizens**

The primary users of this system are general voters, such as citizens in a national or local election, employees in corporate polls, or students participating in campus elections. These individuals require a system that is accessible, secure, and easy to use. By offering a digital interface—either web-based or mobile—the platform allows users to cast their votes remotely, eliminating the need to visit physical polling stations. This feature is especially beneficial for those living in remote areas, persons with disabilities, senior citizens, or citizens residing abroad (NRIs). The system uses identity verification techniques such as OTPs, digital IDs, or biometric authentication to ensure that only eligible users can vote, while cryptographic techniques maintain voter privacy.

2. **Election Authorities / Administrators**

Election officials, government election bodies, institutional administrators, and corporate HR departments fall into this category. These users are responsible for configuring the election settings, registering voters and candidates, overseeing the progress of the election, and declaring results. The blockchain-based system offers them tools to monitor voting activities in real time, receive reports, and ensure that the entire process is conducted in a fair, transparent, and tamper-proof manner. Administrative dashboards and smart contracts reduce the need for manual intervention and improve the efficiency and credibility of elections.

3. Candidates

Candidates who are contesting elections are also important users of the system. They need assurance that the electoral process is unbiased and transparent. Blockchain technology ensures that each vote is recorded immutably, and vote counts are conducted automatically through smart contracts. This prevents manipulation and ensures that candidates and voters alike can trust the system. Candidates can also view their profiles, track voter participation (without accessing personal data), and see verified results once the election ends.

4. **Auditors and Election Observers**

Another crucial group includes independent auditors, election watchdogs, and third-party observers. These users ensure the integrity of the electoral process. The blockchain ledger, which is immutable and publicly accessible, allows for real-time auditing without exposing sensitive information. Observers can independently verify that votes were cast, recorded, and counted correctly, without having to rely on a centralized authority. This level of transparency is a major step forward in addressing issues like vote rigging and post-election disputes.

5. **Developers and Technical Support Teams**

The developers and IT professionals responsible for building, maintaining, and securing the system are also key target users. Their job is to ensure the platform is functioning correctly, handling large-scale traffic during elections, protecting against cyber threats, and integrating features such as authentication, blockchain integration, and smart contracts. They also support system updates, bug fixes, and scalability improvements, especially as the system is adapted for various use cases.

6. **Policy Makers and Government Institutions**

In a broader scope, policymakers and government institutions are secondary users who can evaluate the system for adoption at a national or regional level. They assess its security, transparency, and impact on democratic engagement. A successful implementation can encourage regulatory bodies to support the transition from traditional voting to secure digital voting platforms.

1.2 Problem Statement

Elections are the foundation of a democratic society, where citizens exercise their right to choose their representatives. However, the credibility and efficiency of electoral processes across many regions have come under scrutiny due to recurring issues such as vote tampering, voter fraud, lack of transparency, limited accessibility, and delayed result announcements. Traditional voting systems—whether paper-based or electronic voting machines (EVMs)—are often centralized, making them vulnerable to manipulation, technical glitches, and lack of public trust.

In paper-based voting, challenges such as ballot stuffing, invalid votes, and manual counting errors compromise the integrity of the election. Moreover, results take a significant amount of time to be verified and declared, and recounts or disputes can prolong the process even further. Similarly, electronic voting machines, though faster, still raise concerns about security vulnerabilities, lack of transparency in the voting and counting process, and possible interference.

Another major problem is the **limited accessibility** of traditional voting methods. Voters who live in remote areas, persons with disabilities, elderly citizens, and those living abroad often face difficulty in casting their votes. This leads to reduced voter turnout and questions the inclusiveness of the election process.

In addition, voters do not have the ability to **verify whether their vote was cast, recorded, and counted correctly**. This lack of verifiability fuels mistrust in the system and creates room for misinformation and post-election disputes. In high-stake elections, even a small flaw or suspicion can lead to political unrest or legal battles.

The centralized nature of conventional systems also makes them a **single point of failure**, exposing the entire election infrastructure to internal tampering or cyber-attacks. Moreover, the administrative cost, time, and human resources required to conduct, monitor, and verify elections make the process even more cumbersome.

Therefore, there is an urgent need for a secure, transparent, decentralized, and tamper-proof voting system that addresses these issues. The use of blockchain technology provides an innovative solution to overcome these challenges by offering transparency, immutability, and trust without compromising voter privacy.

This project aims to design and develop a Transparent Voting System to eliminate the problems of traditional systems and ensure fair, secure, and efficient elections for all.

1.2.1 Limitations of Traditional Voting

Traditional voting systems, whether paper-based or electronic, have long served as the backbone of democratic processes. However, as the world evolves technologically, these systems are increasingly being scrutinized for their inefficiencies, vulnerabilities, and lack of transparency.

Several critical limitations hinder the effectiveness of traditional voting methods, thereby impacting the credibility and fairness of elections.

One of the most significant drawbacks of traditional voting is the **lack of transparency**. In paper-based systems, once a vote is cast, it is difficult for the voter to verify if it was accurately counted. The entire process is handled by election officials, and voters must blindly trust that the results have not been manipulated or tampered with. Electronic Voting Machines (EVMs), though faster, also suffer from similar issues. Their internal software and counting mechanisms are often not open for public audit, leading to skepticism about the accuracy and fairness of election outcomes.

Another major limitation is **security**. Paper ballots are vulnerable to physical tampering, ballot box stuffing, and destruction. EVMs, on the other hand, are susceptible to hacking and software manipulation if adequate safeguards are not in place. These security lapses can lead to vote rigging, manipulation of results, and loss of public trust.

Accessibility is also a serious concern. Voters living in remote areas, people with disabilities, elderly citizens, and those residing overseas often find it difficult or impossible to participate in elections due to physical or logistical challenges. Traditional voting requires voters to be physically present at polling stations, which can limit turnout and exclude significant portions of the population.

Additionally, traditional voting systems are **time-consuming and expensive**. The process of printing ballots, setting up polling booths, staffing, security, and manual vote counting demands substantial financial and human resources. Delays in vote tallying can lead to post-election disputes, protests, and even political instability.

Lastly, there is **limited verifiability and auditability** in traditional systems. Voters cannot independently verify whether their vote was recorded and counted correctly. This lack of trust can result in dissatisfaction with the electoral process and low voter engagement.

In conclusion, while traditional voting systems have served democracies for decades, their limitations demand the need for a more secure, transparent, and efficient alternative—one that can be fulfilled by technologies like blockchain.

1.2.2 Challenges in Digital Adoption

The shift towards digital solutions in various sectors, including voting, holds tremendous potential to improve efficiency, security, and accessibility. However, the adoption of digital systems, particularly for something as critical as elections, presents several challenges that need to be addressed for successful implementation. These challenges include technical, social, legal, and infrastructural hurdles that could hinder the widespread acceptance of digital voting systems.

1. **Technological Infrastructure and Accessibility**

One of the primary challenges in the digital adoption of voting systems is the **lack of adequate infrastructure**. In many regions, particularly in developing countries, access to reliable internet services, smartphones, and modern computing devices remains limited. This digital divide can prevent large portions of the population from participating in online elections. Additionally, rural areas may lack the necessary infrastructure to support seamless digital voting. To overcome this barrier, comprehensive digital literacy programs and improvements in internet access must be a part of the transition plan.

2. **Cybersecurity Risks and Data Privacy Concerns**

Digital voting systems are exposed to cybersecurity risks, including hacking, data breaches, and denial-of-service attacks. Ensuring the security of votes and personal information is paramount. If a digital voting platform is compromised, the integrity of the election results is jeopardized. Blockchain technology offers strong encryption and decentralized control, but it is not immune to evolving cyber threats. As a result, creating robust cybersecurity frameworks and ensuring continuous monitoring of digital platforms are essential for fostering trust in digital voting systems.

Moreover, **data privacy** is another significant concern. Voters need assurance that their personal information, including voting choices, will remain private and protected. In traditional voting systems, anonymity is maintained through paper ballots; however, digital voting systems must find innovative ways to safeguard voter privacy, particularly when using technologies like blockchain, where transaction histories are transparent.

3. **Public Trust and Resistance to Change**

Public trust in digital voting systems is a critical barrier. Many voters, election officials, and political entities are accustomed to traditional methods of voting and may be resistant to adopting digital alternatives. Concerns over the **tampering of results**, fraud, and the transparency of the technology contribute to this skepticism. Additionally, voters may feel uneasy about casting votes on unfamiliar platforms and may fear potential technology failures during critical elections.

4. **Legal and Regulatory Issues**

The adoption of digital voting systems requires changes in existing **legal and regulatory frameworks**. Many countries have stringent laws governing elections, which may not be compatible with digital voting technologies. Legal reforms are necessary to ensure that digital voting platforms comply with election laws, particularly with respect to data protection, vote verification, and the integrity of election results.

5. **Cost of Implementation**

Finally, the initial cost of implementing a digital voting system is often high. This includes the development of secure software, purchasing the necessary hardware, training personnel, and ensuring long-term maintenance. For many governments, especially in resource-limited regions, these costs may seem prohibitive despite the long-term benefits.

1.3 Objectives of the Project

The **Transparent Voting System** is designed to address the key challenges and limitations of traditional voting systems while ensuring transparency, security, and efficiency in the electoral process. The core objectives of this project revolve around creating a secure, user-friendly, and decentralized voting system that can be widely adopted in various types of elections. Below are the key objectives of the project:

1. **Enhance Transparency in Voting**

One of the primary objectives is to ensure complete transparency throughout the election process. Traditional voting systems often suffer from a lack of transparency, leading to skepticism and mistrust. By leveraging **blockchain technology**, every vote cast will be recorded as a unique, immutable transaction on a public ledger. This ensures that each vote can be traced and verified independently, eliminating the possibility of tampering or vote manipulation.

2. **Ensure Security and Privacy**

Security is a paramount concern in any voting system. The project aims to create a platform that guarantees the **integrity** and **confidentiality** of voters' choices. Blockchain's cryptographic techniques will prevent any unauthorized access, manipulation, or alteration of votes. Additionally, voter **anonymity** will be maintained through cryptographic methods, ensuring that voters' identities remain private while their votes are securely recorded.

3. **Facilitate Remote Voting and Increase Accessibility**

One of the significant advantages of digital voting is that it allows people to vote remotely, which can significantly increase voter turnout, especially for those who face logistical challenges or physical disabilities. The system will be designed to be accessible via web and mobile platforms, enabling voters to cast their ballots from anywhere, as long as they have internet access. This **remote accessibility** will address the issues faced by voters in rural or remote areas, as well as expatriates living abroad.

4. **Streamline the Election Process with Automation**

The project seeks to reduce the administrative burden associated with traditional elections by automating various processes, including **vote counting, result generation, and audit trails**. The use of **smart contracts** will automate key election functions, such as verifying votes, ensuring the legitimacy of voter identities, and declaring results. This reduces the chances of human error and administrative delays, ensuring faster and more accurate outcomes.

5. **Promote Trust in the Electoral Process**

The project aims to foster public trust in the electoral process by integrating blockchain technology into the voting system. Transparency, security, and verifiability will be core features of the system, allowing voters, political parties, and independent observers to have confidence in the integrity of the election results.

6. **Scalability for Different Election Types**

The system is designed to be scalable and adaptable to a variety of election types, ranging from **local government elections**, **student body elections**, to **corporate voting** and even national elections. The modular nature of the blockchain system will ensure that the platform can scale according to the size and complexity of the election.

1.4 Methodology

The **Transparent Voting System** aims to modernize and secure the electoral process through a decentralized, transparent, and efficient platform. The methodology behind this system involves a series of phases, including requirement gathering, system design, blockchain integration, and testing, each aimed at delivering a functional, secure, and scalable digital voting platform.

1. **Requirement Analysis**

The first step in the methodology is gathering and analyzing the project requirements. This involves understanding the needs of the stakeholders, including voters, election administrators, and auditors. The key requirements are identified, such as ensuring **voter anonymity**, **system security**, **ease of use**, and **real-time vote tallying**. This phase also focuses on the compliance with legal standards and regulations governing elections, ensuring the system meets the criteria for official election purposes.

2. **System Design and Architecture**

Once the requirements are clearly defined, the next phase is to design the architecture of the voting system. The design focuses on scalability, usability, and security. The system is designed with a **decentralized blockchain network**, ensuring that all votes are securely recorded on a public ledger that cannot be tampered with. The system is also designed to support multiple election types, from small organizational elections to large-scale national elections.

The platform will have a **user-friendly interface** for voters, allowing them to securely cast their votes using web or mobile applications. Blockchain integration ensures that each vote is timestamped, cryptographically signed, and stored immutably on the blockchain. The backend is built to support smart contracts for automating the election process, including vote validation and result calculation.

3. **Blockchain Integration**

Blockchain technology forms the backbone of the voting system. The system uses a **private or consortium blockchain** to ensure only authorized nodes (election authorities) participate in validating votes. Blockchain ensures **transparency** by providing an immutable record of each vote cast. **Smart contracts** are used to automate election rules, such as validating voter eligibility and counting votes automatically once the election closes.

4. **Implementation and Development**

The system is developed using modern web technologies for the front-end, such as **ReactJS** or **VueJS**, and back-end technologies like **Node.js** or **Python**. For blockchain integration, platforms like **Ethereum**, **Hyperledger**, or **Solana** can be used based on the desired level of decentralization and scalability. The front-end is designed to be intuitive and responsive, ensuring that all users, regardless of their technical expertise, can easily participate in the election process.

5. **Testing and Evaluation**

After the system is developed, it undergoes rigorous testing. This includes **unit testing**, **integration testing**, and **security testing** to ensure that the system is secure, accurate, and free of vulnerabilities. Blockchain transactions are tested for immutability, and the system is stress-tested to handle high traffic during large elections.

6. **Deployment and Maintenance**

Once the system passes testing, it is deployed on a secure cloud platform or server. Regular maintenance is carried out to ensure system updates, security patches, and scalability are managed effectively.

1.5 Tools & Technologies Used

The **Transparent Voting System** incorporates a variety of cutting-edge technologies to ensure that the election process is secure, transparent, efficient, and user-friendly. The tools and technologies used in this project are selected based on their ability to meet the project's objectives, such as scalability, security, and ease of use.

1. **Blockchain Technology**

The cornerstone of this project is **Blockchain** technology, which ensures **immutability, transparency, and security** of the voting process. Blockchain allows for a decentralized ledger where each vote is recorded as a transaction that cannot be altered once confirmed. For this project, a **private or consortium blockchain** is used, where only authorized participants (election authorities and validators) can access and validate the data. Popular blockchain platforms include:

- **Ethereum**: A decentralized platform for building smart contracts and decentralized applications (dApps). Ethereum's smart contracts ensure that election rules, such as vote validation and counting, are automatically enforced.
- **Hyperledger Fabric**: A permissioned blockchain framework designed for enterprise use, ensuring that only authorized entities can participate in the validation process. It offers enhanced privacy, scalability, and flexibility.
- **Solana**: A high-performance blockchain ideal for handling large-scale transactions at speed, which is beneficial for elections with high voter turnout.

2. **Smart Contracts**

Smart Contracts are self-executing contracts with the terms of the agreement directly written into code. These contracts automatically execute actions based on predefined conditions. In this voting system, smart contracts are used for automating election processes, such as **vote validation**, **voter eligibility checks**, and **automatic vote tallying**. Smart contracts ensure that the election process is transparent, tamper-proof, and executed without human intervention, reducing errors and fraud.

3. **Frontend Technologies**

To provide an intuitive and responsive user interface for voters, the system uses modern **frontend technologies**:

- **ReactJS**: A JavaScript library for building user interfaces, which helps create fast and interactive UI components for the voting platform.
- **Tailwind CSS**: A utility-first CSS framework that allows for quick styling of the application, ensuring a clean and responsive design that works across different devices.
- **HTML5 & CSS3**: Standard web technologies used for structuring and styling the web-based voting platform.

4. **Backend Technologies**

For the backend, the system uses server-side technologies to handle data processing, vote validation, and interactions with the blockchain network:

- **Node.js**: A JavaScript runtime environment that enables efficient, scalable server-side applications. It is ideal for handling multiple requests concurrently, making it perfect for managing large-scale elections.
- **Express.js**: A minimalistic framework for building robust APIs, used to handle HTTP requests and responses between the frontend and backend of the voting platform.

5. **Database Technologies**

For temporary data storage (e.g., user data, election metadata), **MongoDB** is used. MongoDB is a NoSQL database that offers scalability, flexibility, and performance to handle large amounts of unstructured data.

6. **Cloud Hosting & Deployment**

The system is deployed on secure cloud platforms like **AWS**, **Google Cloud**, or **Azure** for high availability, scalability, and security. Cloud hosting ensures that the voting system can handle high volumes of traffic and is accessible from anywhere.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study is an assessment of the practicality of a proposed project or system. It aims to objectively and rationally uncover the strengths and weaknesses of the system, the opportunities it presents, the resources required to carry it through, and ultimately the prospects for success. For the **"Transparent Voting System "**, a comprehensive feasibility study is conducted in five key areas: technical, economic, legal, operational, and scheduling feasibility.

2.1 Technical Feasibility

Technical feasibility assesses whether the current technology is capable of meeting the requirements of the system. In the case of a blockchain-based voting system, the technology stack primarily involves:

- **Blockchain Network:** To maintain a distributed and tamper-proof ledger of votes.
- **Smart Contracts:** To automate the logic of voting, such as voter verification, vote casting, and result declaration.
- **Cryptographic Techniques:** For ensuring data privacy, encryption of votes, and digital signatures for identity validation.
- **Web and Mobile Interface:** For providing a user-friendly platform for voters to interact with the system.

Blockchain technology has already proven its success in domains like cryptocurrency, supply chain, and secure transactions. Its immutability and decentralized nature ensure that once a vote is recorded, it cannot be altered, making it ideal for a voting system.

Additionally, smart contract platforms like **Ethereum**, **Hyperledger Fabric**, and **Polygon** support the secure and decentralized execution of business logic. Therefore, the proposed system is technically feasible and can be implemented using existing frameworks and technologies.

2.2 Economic Feasibility

Economic feasibility evaluates whether the project is financially viable. Although the initial implementation of blockchain-based voting systems can be costly, it offers long-term cost-saving benefits.

Initial Costs:

- Infrastructure setup (blockchain nodes, servers)
- Software development (UI, smart contracts, backend)
- Cybersecurity enhancements
- Voter education and training programs

Long-Term Benefits:

- Reduced costs associated with physical polling booths, paper ballots, and manual counting
- Lower manpower requirement
- Elimination of fraud-related costs and re-elections
- High scalability with minimal additional cost

With time, the system becomes **cost-efficient**, particularly during national or state-wide elections, where expenses are considerably high. Hence, despite high initial investment, the **Return on Investment (ROI)** over multiple election cycles justifies the project's economic feasibility.

2.3 Legal Feasibility

Legal feasibility refers to whether the system complies with the legal framework governing elections and data protection.

The proposed blockchain voting system must adhere to:

- **Indian IT Act, 2000**
- **Data Protection Bill (India)**
- **Election Commission of India's guidelines**
- **Right to Privacy under Article 21 of the Indian Constitution**

The system must include robust mechanisms for:

- **Voter identity verification** using Aadhaar, Voter ID, or biometrics.
- **Consent-based data processing**, as per privacy laws.
- **Auditability** of elections without compromising vote anonymity.

Legal feasibility will also involve coordination with regulatory authorities and may require formal approval for nationwide implementation. With proper encryption, transparent auditing features, and compliance with cyber laws, the project is legally viable.

2.4 Operational Feasibility

Operational feasibility determines whether the system can function effectively in the real world and meet user expectations.

Compatibility:

- The system is designed to **integrate with existing voter databases** maintained by the Election Commission.
- Voting operations such as result counting, user registration, and verification are automated via smart contracts.

Usability:

- The UI is simple and accessible through both web and mobile interfaces.
- Support for multiple languages ensures accessibility to rural and urban populations alike.

Stakeholder Readiness:

- Requires **training sessions** for government officials and electoral staff.
- Awareness campaigns are needed to build public trust in the technology.

The solution provides **greater transparency**, **real-time auditability**, and **instant result declaration**, which improves public confidence in the electoral process.

2.5 Schedule Feasibility

Schedule feasibility assesses whether the system can be developed and deployed within a practical and acceptable time frame. For this project, which has a target duration of approximately **3 months**, the development approach is carefully planned to ensure delivery within the stipulated timeline using **Agile development methodology** with overlapping tasks and focused sprints.

The compressed project timeline is as follows:

Phase	Duration	Activities
Week 1–2 (Planning)	2 Weeks	Finalizing requirements, creating use-case diagrams, and defining scope.
Week 3 (Design)	1 Week	Designing UI mockups, database schema, system architecture (including blockchain setup).
Week 4–7 (Development)	4 Weeks	Coding frontend (voter portal, admin panel), backend (authentication, blockchain integration), and smart contracts.
Week 8 (Testing)	1 Week	Unit testing, integration testing, validation of vote flow & data integrity.

Phase	Duration	Activities
Week 9–10 (Deployment)	2 Weeks	Hosting the system on server/local network, conducting user demo, and admin training.
Week 11–12 (Review & Documentation)	2 Weeks	Running mock elections, collecting feedback, and preparing final documentation/report.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 System Requirement Specification

The Transparent Voting System is a secure, blockchain-integrated platform developed using **PHP** and **MySQL**. It aims to digitize the voting process. -like structures and cryptographic techniques, it ensures voter privacy, vote integrity, and tamper-proof recording.

3.2 Specific Requirements

1. User authentication system for **voters and admins**.
2. Voters should be able to **vote once** and receive confirmation.
3. Votes are recorded securely and **cannot be modified** post-submission.
4. Voting records must be **visible to authorized personnel** for transparency.
5. **Admin dashboard** to manage elections, users, and results.
6. Use of **hashing techniques** (e.g., SHA256) to ensure vote integrity.
7. Simulated or real-time **blockchain-like implementation** for recording votes immutably.

3.3 Hardware Specification

Component	Minimum Requirement
Processor	Intel i3 / AMD equivalent or above
RAM	4 GB (8 GB recommended)
Hard Disk	100 GB HDD or SSD
Devices Supported	PC, Laptop (Smartphone for web access)
Internet	Stable connection (5 Mbps or higher)

3.4 Software Specification

Software Component	Specification
Operating System	Windows, Linux, or macOS
Backend Technology	PHP (v7.4 or higher)
Frontend Technology	HTML, CSS, JavaScript
Database	MySQL
Web Server	Apache (XAMPP/LAMP stack)
Blockchain Simulation	PHP-based logic or third-party API
Libraries/Tools	phpMyAdmin, SHA256 for hashing
Version Control	Git, GitHub

3.5 Functional Requirements

1. **User Registration/Login**
   - Separate portals for voters and admins.
2. **Voter Identity Verification**
   - Admin registers eligible users; optional OTP/email confirmation for login.
3. **Vote Casting**
   - Voters select and submit their vote once per election.
4. **Vote Hashing and Storage**
   - Each vote is hashed (e.g., using SHA256) and stored in a blockchain-style chain (linked via hashes in MySQL).
5. **Vote Confirmation**
   - Voter receives confirmation and can verify that the vote was recorded.
6. **Admin Features**
   - Add/view elections, manage users, and monitor results.

7. **Result Calculation**
    - The system auto-calculates results once voting ends.

3.6 Non-Functional Requirements

1. **Security**
    - Password encryption, vote hashing, secure sessions.
2. **Transparency**
    - Publicly visible but anonymized results via admin dashboard.
3. **Usability**
    - Clean UI/UX for both voters and admins.
4. **Maintainability**
    - Modular PHP scripts and use of comments/documentation.
5. **Availability**
    - System must be up and accessible during election periods.
6. **Portability**
    - Can run on any system with a LAMP/XAMPP setup.

3.7 Performance Requirements

1. **Vote Recording Speed**
    - Vote entries and hashing should complete in under 1 second.
2. **Concurrent Users**
    - System should support at least 300–500 users simultaneously.
3. **Response Time**
    - Page loading and interaction responses under 2 seconds.
4. **Database Handling**
    - MySQL should handle real-time insertions and read queries without lags.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

The existing voting systems used today are mostly centralized in nature and include traditional paper-based voting or basic web-based electronic voting platforms. These systems often suffer from various limitations in terms of transparency, security, and efficiency.

Features of the existing system:

- Centralized control over the entire voting process.
- Vote data is stored in traditional databases that can be vulnerable to manipulation or unauthorized access.
- Voters have no way to verify whether their vote has been correctly recorded or counted.
- Results are often calculated manually or semi-automatically, leading to delays.
- Limited access for people in remote areas or those with physical disabilities.

Drawbacks:

- Risk of tampering or alteration of votes by internal or external attackers.
- Lack of transparency and accountability in vote handling and counting.
- Delayed result announcement due to manual intervention.
- No real-time confirmation or verification of vote submission for voters.
- Single point of failure due to centralized server/database architecture.

4.2 Proposed System

The proposed system is a Transparent Voting System based on blockchain principles, developed using PHP and MySQL. It addresses the challenges of the existing system by providing a secure, transparent, and verifiable online voting platform. The system simulates blockchain features such as immutability and decentralization using hash linking of records in the MySQL database.

Features of the proposed system:

- Each vote is hashed using a cryptographic function (like SHA256) and linked to the previous vote, forming a blockchain-like structure.
- Voters can only vote once, and their identity is kept private while ensuring accountability.
- Votes cannot be modified or deleted once recorded.
- Voters receive confirmation that their vote has been successfully submitted and stored.
- Admins can manage elections, but they cannot alter vote data due to hash verification.
- The system is accessible via web browsers, increasing convenience and participation.

Advantages over the existing system:

- Ensures vote integrity and prevents tampering.
- Increases transparency and builds trust in the electoral process.
- Provides real-time verification and confirmation to voters.
- Reduces manual effort and speeds up result calculation through automated PHP scripts.
- Eliminates dependency on a single authority, reducing the risk of internal manipulation.
- Can be deployed for various scales of elections, from the institutional to the national level.

CHAPTER 5

SYSTEM DESIGN

5.1 Project Modules

1. **Voter Registration Module**
   This module handles the secure registration of voters. It verifies the identity and eligibility of individuals before allowing them to vote. Each voter is assigned unique credentials to ensure one-person-one-vote.
2. **Election Management Module**
   Admins use this module to create and manage elections. It includes functionalities such as setting election timelines, defining candidates, configuring voting rules, and monitoring ongoing elections.
3. **Result Tallying and Reporting Module**
   After voting ends, this module automatically counts the votes stored in the blockchain-style database and displays the results. It ensures that the tallying is tamper-proof and fully transparent to build public trust.
4. **Voter Interface Module**
   This module provides a secure and user-friendly interface for voters. They can log in, view active elections, cast their vote, and receive a confirmation that their vote has been successfully recorded.

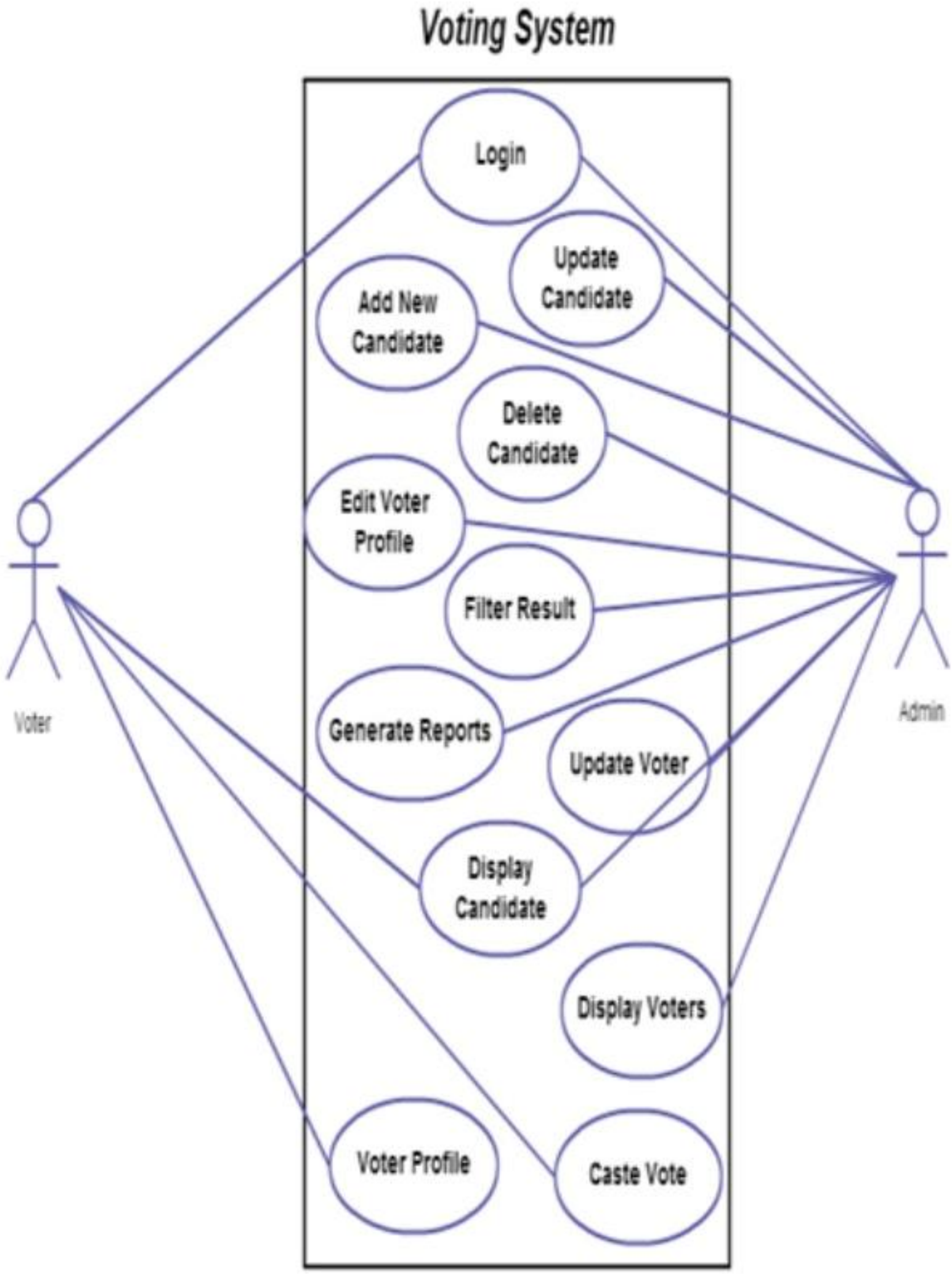
5.2 Use-Case Diagram

Actors:

- Admin
- Voter

Use-Cases:

- Admin: Login, Create Election, Add Candidates, Register Voters, View Results
- Voter: Register/Login, View Election List, Cast Vote, Verify Vote


Voting System
Login
Update Candidate
Add New Candidate
Delete Candidate
Edit Voter Profile
Filter Result
Generate Reports
Update Voter
Display Candidate
Display Voters
Voter Profile
Caste Vote
Voter
Admin

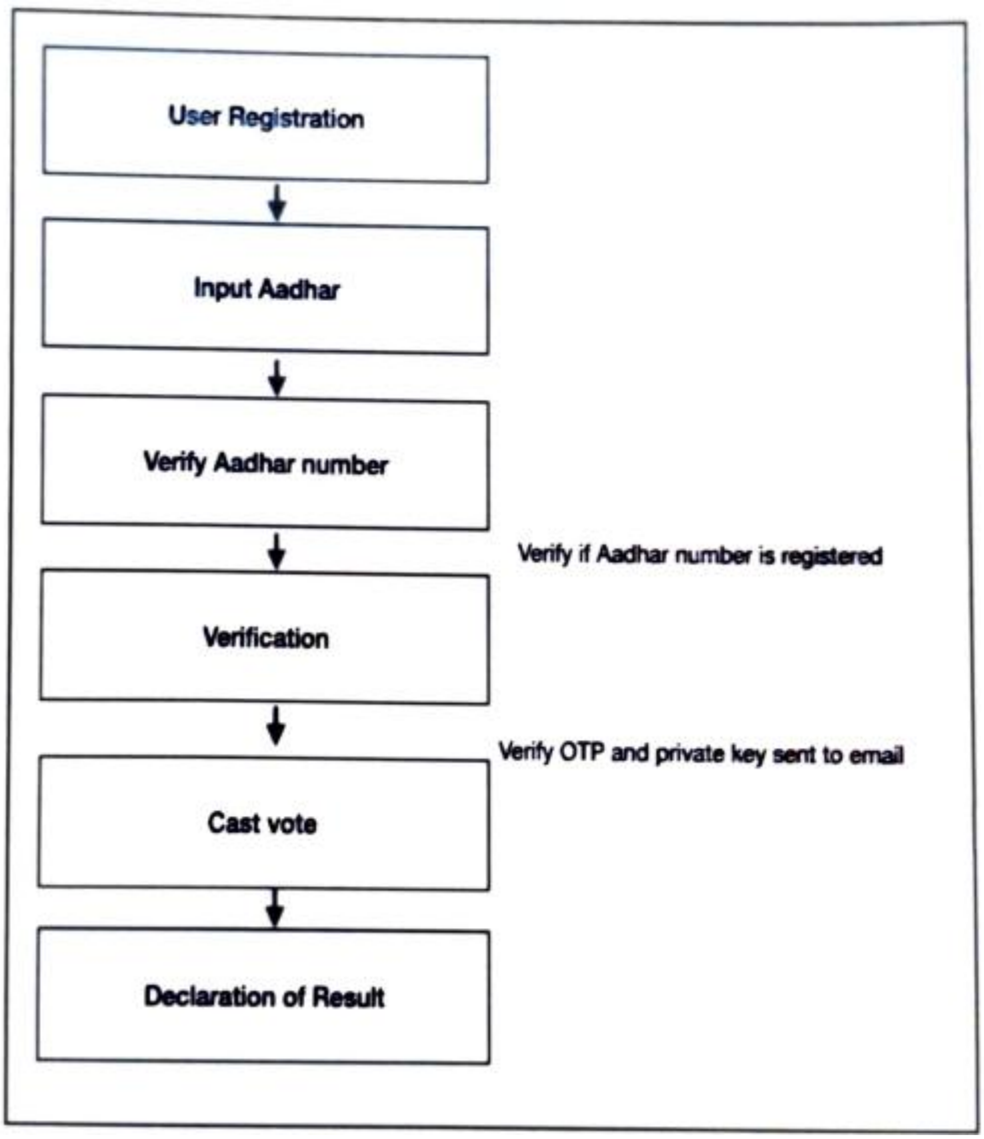
5.3 Data Flow Diagram (Level 0)

The Level 0 DFD shows the overall system interaction between external entities (Voter/Admin) and the core processes. It includes:

- Voter providing input like Aadhaar
- System verifying and allowing voting
- Admin managing elections and declaring results

Main Processes:

- Voter Registration
- Authentication & Verification
- Vote Casting
- Result Declaration



5.3.1 Level 1 Data Flow Diagram

The Level-1 Data Flow Diagram (DFD) of the Transparent Voting System breaks down the initial phase of the voting process, focusing primarily on **user registration and Aadhaar verification**. This level provides a more detailed look into the input validation and authorization process.

Processes Involved:

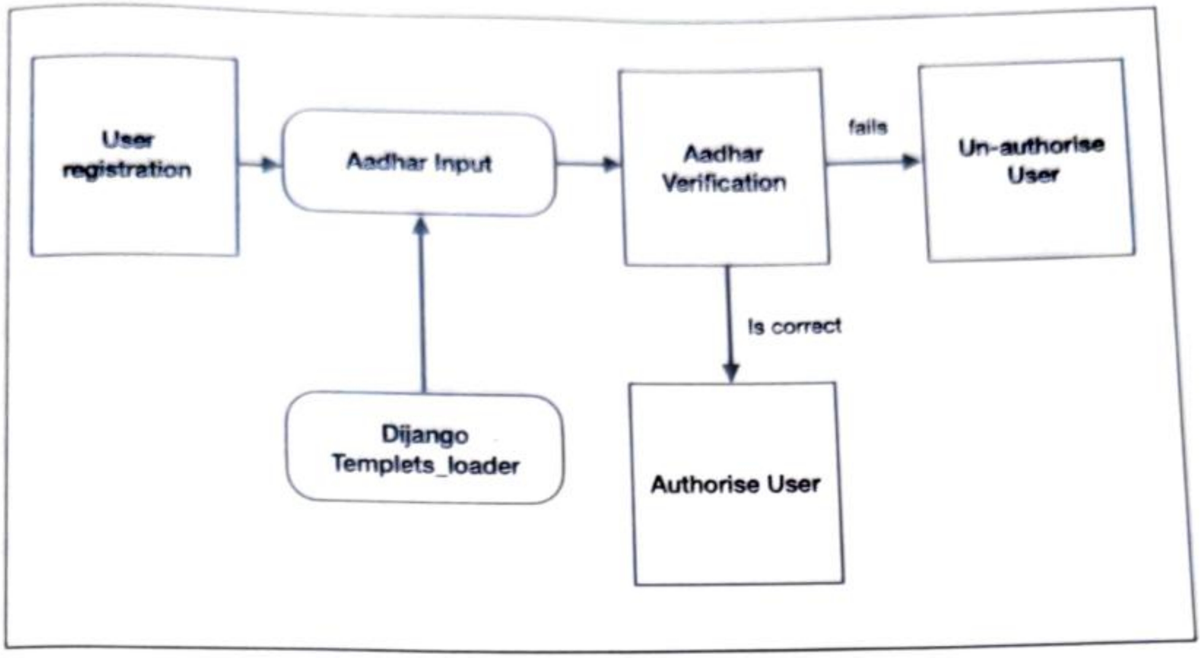
1. **User Registration:**
   - The voter begins the process by initiating registration on the system interface.
   - Basic details are collected, and the user proceeds to the Aadhaar input stage.
2. **Aadhaar Input:**
   - The user inputs their Aadhaar number through the interface.
   - The system handles this input, likely managed through a front-end framework and template loader (as referenced in the diagram by Django Templet_loader though your project uses PHP, the logic is similar).
3. **Aadhaar Verification:**
   - The system verifies the Aadhaar number with the database.
   - This step ensures that the Aadhaar number is valid, registered, and matches the user's record.
4. **Authorization Decision:**
   - If the Aadhaar verification is **successful**, the user is **authorized** and allowed to proceed to the voting module.
   - If the verification **fails**, the system marks the individual as an **unauthorized user** and blocks access to further processes.

Supporting Component:

- **Template Loader (Django Templet_loader):**
  - While the label in the diagram says Django, in your PHP-based system this would correspond to your **HTML/PHP templates or front-end view engine** that renders pages such as the Aadhaar input form.

Data Stores & Flow:

- Inputs and outputs flow between the user and system processes.
- The system manages user status (authorized or unauthorized) based on the verification result.
- Data consistency and secure handling are essential at this level to prevent fraud or misuse.



5.3.2 Level-2 Dataflow Diagram

The Level-2 Data Flow Diagram (DFD) dives deeper into the technical working of the voting system after the user has been authenticated. It focuses on how votes are securely cast and stored principles like hashing and block creation. This level highlights the security mechanisms integrated into the system to maintain data integrity and ensure voter anonymity.

1. Authorise User

- This process starts with the user who has successfully passed Aadhaar verification in the previous level.
- Only authorized users are allowed to proceed to voting-related operations.

2. Generate OTP

- An OTP (One-Time Password) is generated for secure user verification.
- This OTP is sent to the user via **SMTP protocols** to their registered email address.

- It adds a second layer of security and prevents unauthorized voting.

3. SMTP Protocols

- Simple Mail Transfer Protocol (SMTP) is used to deliver the OTP securely to the user's email.
- This ensures only the rightful user, with access to their registered email, can proceed further.

4. Create Blocks and Generate Hash Values

- Once the OTP is verified, the system proceeds to vote storage **concepts**.
- Votes are encrypted and stored in blocks.
- Each vote is hashed using **SHA-256** and **Merkle Tools**, ensuring:
    - **Tamper-proof** data
    - **Integrity** of voting records
    - **Non-repudiation** of cast votes

5. Merkel Tools and SHA-256

- These tools are used to:
    - Convert votes into cryptographic hash values.
    - Create a chain of blocks where each block depends on the hash of the previous one.
- This provides **transparency**, **immutability**, and **trust** in the system.

6. Cast Vote

- After successful OTP verification and hash generation, the user casts their vote.
- The vote is securely recorded and linked in the blockchain structure.

7. Declare Result

- Once voting concludes, the results are calculated based on the blocks.
- Since the blocks are immutable, the results are trustworthy and transparent.
- The final election outcome is then declared to the public.

CHAPTER 6

IMPLEMENTATION

This chapter describes the practical realization of the Transparent Voting System technology. It explains the core idea, the algorithms used to ensure transparency and security, and how the project is divided into functional modules to handle various responsibilities effectively.

6.1 Concept

The Transparent Voting System is developed with the primary aim to ensure **secure, tamper-proof, and transparent elections technology**. The concept focuses on:

- **Decentralization:** Eliminating a single point of failure by storing votes on a distributed ledger.
- **Transparency:** All transactions (votes) are recorded and visible to all participants.
- **Immutability:** Once a vote is cast and recorded, it cannot be altered or deleted.
- **Security:** Each vote is encrypted and securely linked with blockchain nodes.
- **Anonymity:** Voter identity is protected while ensuring one-person-one-vote integrity.

The system uses a combination of frontend (user interface for voters and admins), backend (PHP server logic), MySQL database (for user records), and a **blockchain simulation** (for recording votes in a block structure).

6.2 Algorithms

The algorithms used in this project are central to its security and functionality:

a. **Blockchain Algorithm (Simplified Version)**

- **Block Structure**: Each block contains a vote, timestamp, hash of the previous block, and its own hash.
- **Hash Function**: SHA-256 is used to generate a unique fingerprint for each block.

- **Chaining**: Every block is linked to the previous block using its hash to maintain immutability.
- **Validation**: During each vote submission, the system verifies the blockchain's integrity to prevent tampering.

b. **Voting Verification Algorithm**

- Ensures that:
  - Voter is registered and authenticated.
  - Voter has not voted before (using a voting status flag in the database).
  - Vote is properly encrypted and inserted into the blockchain structure.

c. **Admin Results Calculation**

- Parses the blockchain to count votes.
- Tally results securely without accessing individual voter data.
- Uses simple counting logic from blockchain entries to generate outcome.

6.3 Functional Modules

The system is divided into four key functional modules:

1. Voter Registration Module

- Allows users to sign up with personal details.
- Admin verifies and activates accounts.
- Ensures only eligible voters can access the voting feature.

2. Voting Module

- Authenticated users can cast their vote.
- The vote is encrypted and added to the blockchain.
- Once voted, the user cannot vote again (status updated).

3. Blockchain Module

- Handles the creation and management of blocks.
- Maintains the integrity of the chain by ensuring the correct linking of hashes.
- Allows verification of vote data for transparency.

4. Admin Panel

- View and manage registered voters.
- Initiate, monitor, or stop voting process.
- Access the result analysis by parsing blockchain entries.
- Ensures administrative control with transparency and audit logs.

CHAPTER 7

Code

Admin > index

```
<?php

	session_start();

	if(isset($_SESSION['admin'])){

	header('location:home.php');

	}

?>

<?php include 'includes/header.php'; ?>

<body class="hold-transition login-page">

<div class="login-box">

	<div class="login-logo">

		<b>Voting System</b>

	</div>



	<div class="login-box-body">

	<p class="login-box-msg">Sign in to start your session</p>



	<form action="login.php" method="POST">

		<div class="form-group has-feedback">
```

```
                <input type="text" class="form-control" name="username"
placeholder="Username" required>

                <span class="glyphicon glyphicon-user form-control-
feedback"></span>

                </div>

        <div class="form-group has-feedback">

          <input type="password" class="form-control" name="password"
placeholder="Password" required>

          <span class="glyphicon glyphicon-lock form-control-
feedback"></span>

        </div>

                <div class="row">

                        <div class="col-xs-4">

                        <button type="submit" class="btn btn-primary btn-block
btn-flat" name="login"><i class="fa fa-sign-in"></i> Sign In</button>

                </div>

                </div>

        </form>

        </div>

        <?php

                if(isset($_SESSION['error'])){

                        echo "

                                <div class='callout callout-danger text-center mt20'>

                                        <p>".$_SESSION['error']."</p>

                                </div>
```

```
                    ";
                    unset($_SESSION['error']);
                }
        ?>
</div>
<?php include 'includes/scripts.php' ?>
</body>
</html>
```

Login Home file

```
<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue layout-top-nav">
<div class="wrapper">

<?php include 'includes/navbar.php'; ?>

<div class="content-wrapper">
<div class="container">

<!-- Main content -->
<section class="content">
<?php
$parse = parse_ini_file('admin/config.ini', FALSE, INI_SCANNER_RAW);
```

```
$title = $parse['election_title'];

?>

<h1 class="page-header text-center title"><b><?php echo strtoupper($title);
?></b></h1>

<div class="row">

<div class="col-sm-10 col-sm-offset-1">

<?php

if(isset($_SESSION['error'])){

?>

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>

<ul>

<?php

foreach($_SESSION['error'] as $error){

echo "

<li>".$error."</li>

";

}

?>

</ul>

</div>

<?php

unset($_SESSION['error']);
```

```
}

if(isset($_SESSION['success'])){

echo "

<div class='alert alert-success alert-dismissible'>

<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>

<h4><i class='icon fa fa-check'></i> Success!</h4>

".$_SESSION['success']."

</div>

";

unset($_SESSION['success']);

}


?>


<div class="alert alert-danger alert-dismissible" id="alert" style="display:none;">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

<span class="message"></span>

</div>


<?php
```

```
$sql = "SELECT * FROM votes WHERE voters_id = '".$voter['id']."'";

$vquery = $conn->query($sql);

if($vquery->num_rows > 0){

?>

<div class="text-center">

<h3>You have already voted for this election.</h3>

<a href="#view" data-toggle="modal" class="btn btn-flat btn-primary btn-
lg">View Ballot</a>

</div>

<?php

}

else{

?>

<!-- Voting Ballot -->

<form method="POST" id="ballotForm" action="submit_ballot.php">

<?php

include 'includes/slugify.php';


$candidate = '';

$sql = "SELECT * FROM positions ORDER BY priority ASC";

$query = $conn->query($sql);

while($row = $query->fetch_assoc()){

$sql = "SELECT * FROM candidates WHERE position_id='".$row['id']."'";
```

```
$cquery = $conn->query($sql);

while($crow = $cquery->fetch_assoc()){

$slug = slugify($row['description']);

$checked = '';

if(isset($_SESSION['post'][$slug])){

$value = $_SESSION['post'][$slug];


if(is_array($value)){

foreach($value as $val){

if($val == $crow['id']){

$checked = 'checked';

}

}

}

else{

if($value == $crow['id']){

$checked = 'checked';

}

}

}

$input = ($row['max_vote'] > 1) ? '<input type="checkbox" class="flat-red
'.$slug.'" name="'.$slug."[]".'" value="'.$crow['id'].'" '.$checked.'>' : '<input
type="radio" class="flat-red '.$slug.'"
name="'.slugify($row['description']).'" value="'.$crow['id'].'" '.$checked.'>';
```

```
$image = (!empty($crow['photo'])) ? 'images/'.$crow['photo'] :
'images/profile.jpg';

$candidate .= '

<li>

'.$input.'<button type="button" class="btn btn-primary btn-sm btn-flat clist
platform" data-platform="'.$crow['platform'].'" data-
fullname="'.$crow['firstname'].' '.$crow['lastname'].'"><i class="fa fa-
search"></i> Platform</button><img src="'.$image.'" height="100px"
width="100px" class="clist"><span class="cname
clist">'.$crow['firstname'].' '.$crow['lastname'].'</span>

</li>

';

}



$instruct = ($row['max_vote'] > 1) ? 'You may select up to
'.$row['max_vote'].' candidates' : 'Select only one candidate';



echo '

<div class="row">

<div class="col-xs-12">

<div class="box box-solid" id="'.$row['id'].'">

<div class="box-header with-border">

<h3 class="box-title"><b>'.$row['description'].'</b></h3>

</div>

<div class="box-body">

<p>'.$instruct.'
```

```
<span class="pull-right">

<button type="button" class="btn btn-success btn-sm btn-flat reset" data-desc="'.slugify($row['description']).'"><i class="fa fa-refresh"></i> Reset</button>

</span>

</p>

<div id="candidate_list">

<ul>

'.$candidate.'

</ul>

</div>

</div>

</div>

</div>

</div>

';


$candidate = '';


}


?>

<div class="text-center">
```

```
<button type="button" class="btn btn-success btn-flat" id="preview"><i class="fa fa-file-text"></i> Preview</button>

<button type="submit" class="btn btn-primary btn-flat" name="vote"><i class="fa fa-check-square-o"></i> Submit</button>

</div>

</form>

<!-- End Voting Ballot -->

<?php

}


?>


</div>

</div>

</section>


</div>

</div>


<?php include 'includes/footer.php'; ?>

<?php include 'includes/ballot_modal.php'; ?>

</div>


<?php include 'includes/scripts.php'; ?>
```

```
<script>

$(function(){

$('.content').iCheck({

checkboxClass: 'icheckbox_flat-green',

radioClass: 'iradio_flat-green'

});


$(document).on('click', '.reset', function(e){

e.preventDefault();

var desc = $(this).data('desc');

$('.'+desc).iCheck('uncheck');

});


$(document).on('click', '.platform', function(e){

e.preventDefault();

$('#platform').modal('show');

var platform = $(this).data('platform');

var fullname = $(this).data('fullname');

$('.candidate').html(fullname);

$('#plat_view').html(platform);

});


$('#preview').click(function(e){
```

```
e.preventDefault();

var form = $('#ballotForm').serialize();

if(form == ''){

$('.message').html('You must vote atleast one candidate');

$('#alert').show();

}

else{

$.ajax({

type: 'POST',

url: 'preview.php',

data: form,

dataType: 'json',

success: function(response){

if(response.error){

var errmsg = '';

var messages = response.message;

for (i in messages) {

errmsg += messages[i];

}

$('.message').html(errmsg);

$('#alert').show();

}

else{
```

```
$('#preview_modal').modal('show');

$('#preview_body').html(response.list);

}

}

});

}


});


});

</script>

</body>

</html>
```

Chapter 8

Result

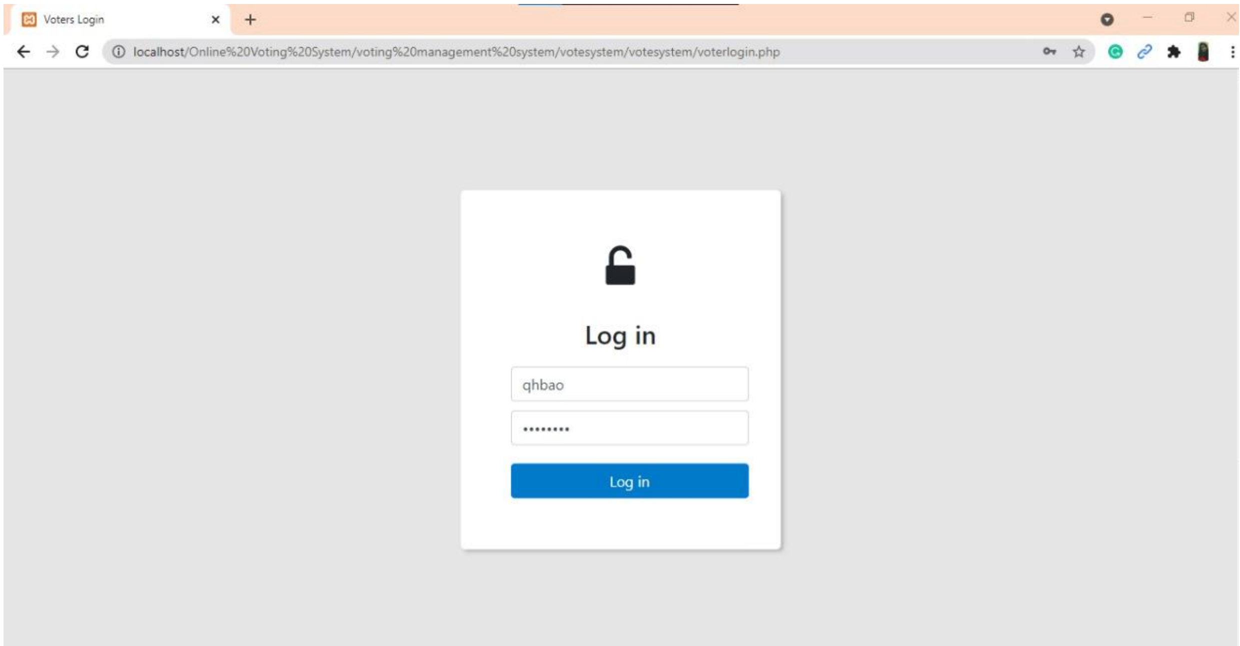


Fig 8.1 Login Page

This login page verifies voter identity using username and password before accessing the voting system, ensuring secure authentication. It’s part of the blockchain-based system ensuring transparency and tamper-proof voting.

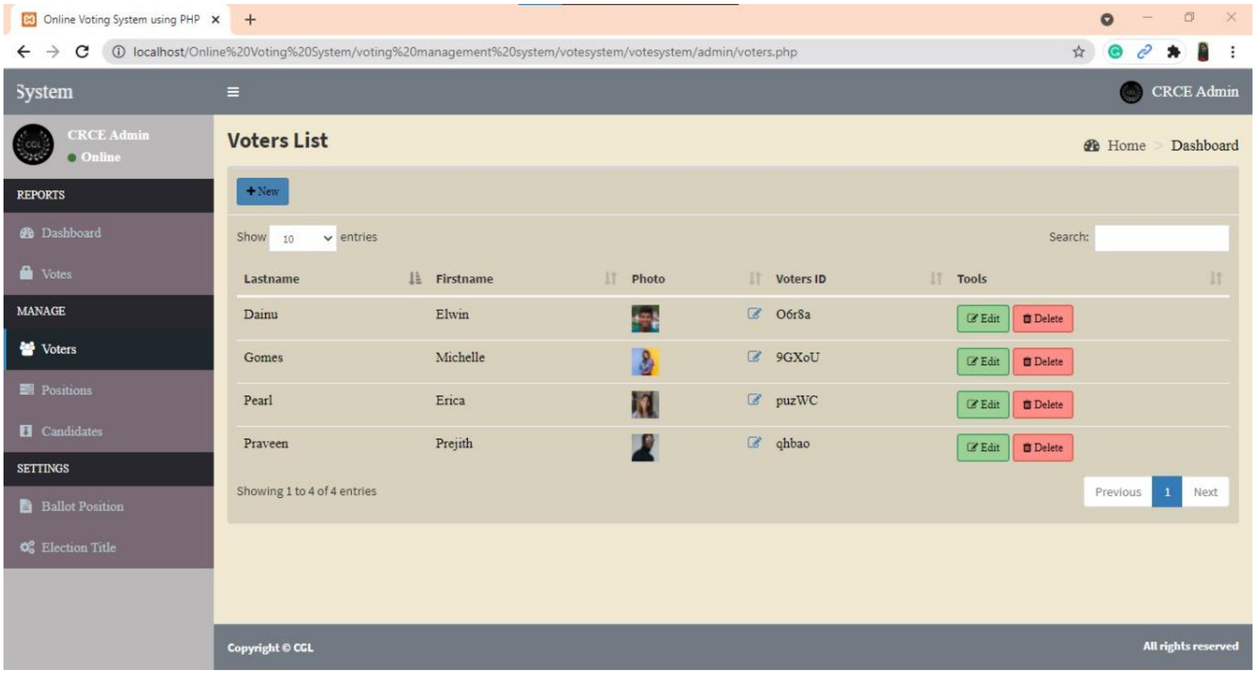


Fig 8.2 Voters List

The Voter List interface displays all registered voters with their last name, first name, photo, and unique voter ID. Admins can manage entries using "Edit" and "Delete" options. This ensures accurate voter data handling and secure authentication within the blockchain-based online voting system, promoting transparency and organized election management.

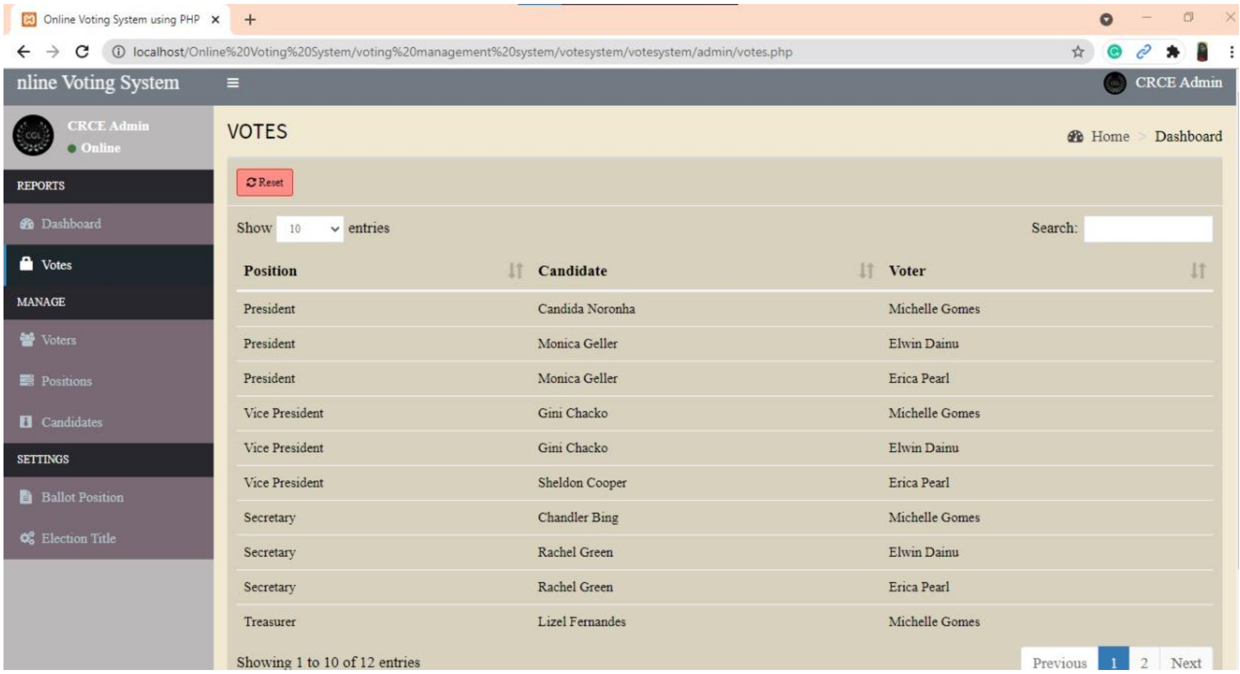


Fig 8.3 Votes

The Votes list displays recorded votes with details like position, candidate name, and the voter who cast the vote. It ensures transparency and traceability in the election process within a secure online voting system. Admins can monitor voting activity, verify participation, and maintain integrity in a blockchain-enabled electoral environment.

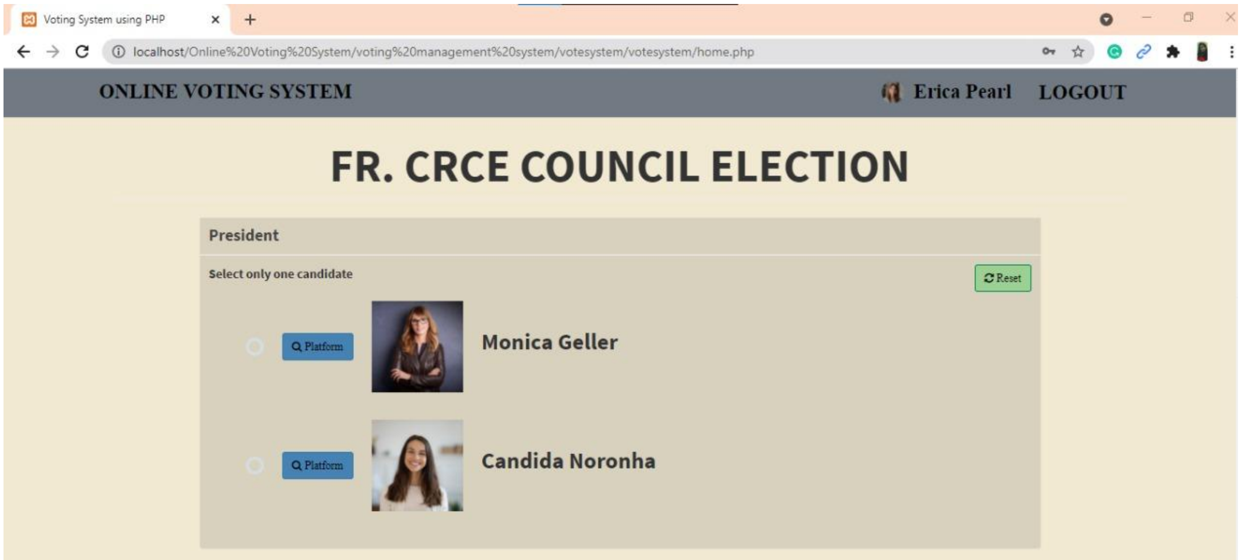


Fig 8.4 FR. CRCE COUNCIL ELECTION

This page displays candidates for the President position in the FR. CRCE Council Election. Voters can view each candidate's photo, name, and platform before selecting one. The system ensures transparency, fairness, and simplicity by allowing only one vote per user, making the election process easy and secure for all participants.

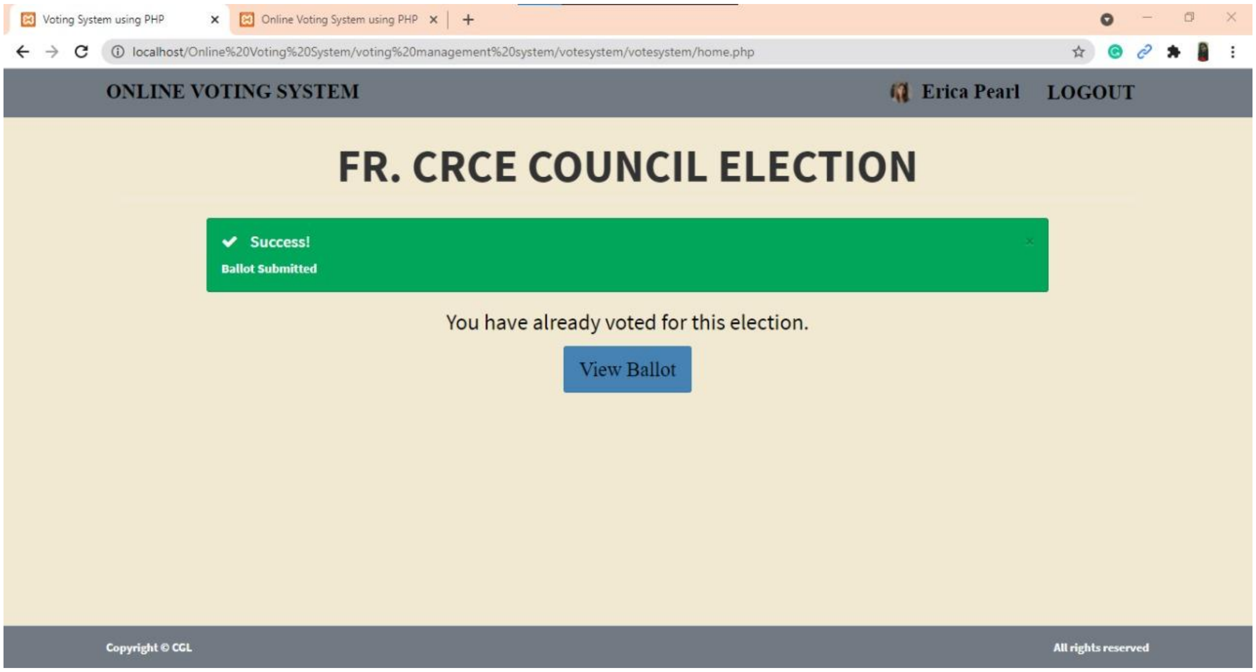


Fig 8.5 Vote Registered Successfully

This screen confirms the voter's ballot has been successfully submitted in the FR. CRCE Council Election. A green success message appears, and the system prevents duplicate voting by showing "You have already voted" with an option to view the submitted ballot.

CHAPTER 9

TESTING

Testing is a critical phase in software development to verify that the system performs as expected, is free from major bugs, and meets user requirements. For the Transparent Voting System, a combination of testing methods was used to ensure **functionality, security, and user satisfaction**.

9.1 Methods of Testing

9.1.1 Unit Testing

- **Purpose**: To test individual components or functions of the system in isolation.
- **Examples**:
  - Testing the createBlock() function to ensure it correctly generates hash and links to the previous block.
  - Verifying form validations on the voter registration and login pages.
  - Testing the database insertion function for votes.

9.1.2 Integration Testing

- **Purpose**: To test how different modules work together.
- **Examples**:
  - Testing interaction between the frontend voting form and backend PHP script that inserts the vote.
  - Ensuring blockchain logic correctly receives and processes data from the voting module.
  - Verifying voter status update after a vote is cast.

9.1.3 Validation Testing

- **Purpose**: To ensure the system meets the defined requirements and behaves correctly with both valid and invalid data.
- **Examples**:

- Entering invalid login credentials and checking error messages.
- Submitting an empty vote form to check required field validation.
- Trying to vote twice with the same user.

9.1.4 User Acceptance Testing (UAT)

- **Purpose**: To validate that the system works as expected from the user's perspective.
- **Method**:
  - Testers (students/teachers/admins) used the system in a real or simulated environment.
  - Collected feedback to improve UI and fix usability issues.
- **Results**:
  - System was found to be user-friendly.
  - Admin dashboard was functional and intuitive.
  - Voting flow was clear and transparent.

9.2 Test Cases

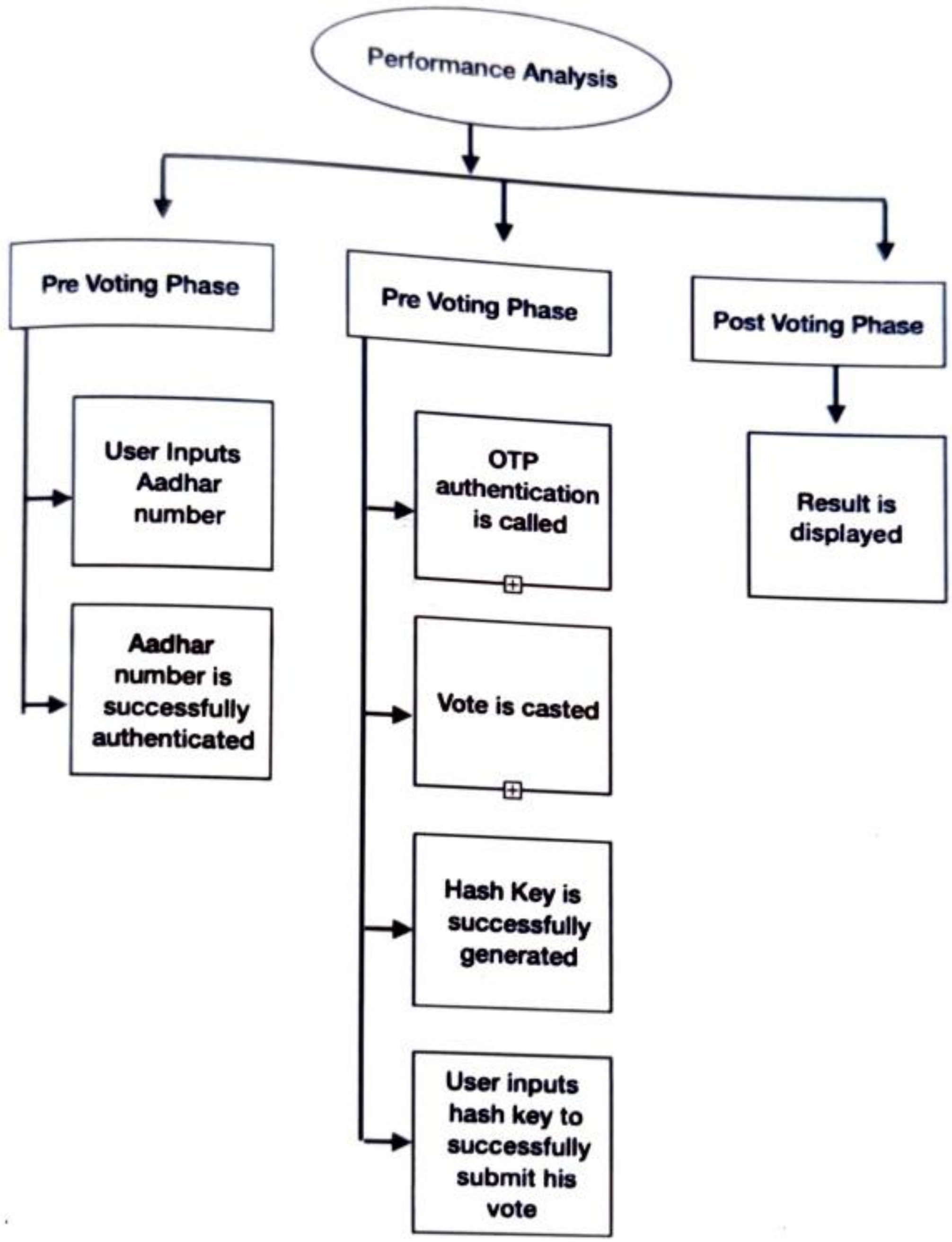
Test Case ID	Description	Input	Expected Output	Status
TC01	Voter Login with correct credentials	Email: valid, Password: valid	Dashboard loads successfully	Pass
TC02	Voter Login with incorrect credentials	Email: valid, Password: wrong	Error message displayed	Pass
TC03	Double voting attempt	Same user tries to vote twice	System blocks second attempt	Pass
TC04	Admin login and view results	Admin credentials	Admin dashboard with live vote count	Pass
TC05	Vote submission blockchain check	Voter submits vote	Vote stored as a new block in the chain	Pass

Test Case ID	Description	Input	Expected Output	Status
TC06	Registration with invalid email	Invalid email format	"Invalid email" error message	Pass
TC07	Vote without login	Direct URL access to voting page	Redirected to login page	Pass
TC08	Result tally integrity	Blockchain parsed for results	Accurate vote counts displayed	Pass
TC09	Blockchain tamper check	Modify a block manually (test)	System flags tampering or invalid chain detected	Pass
TC10	Database connection failure simulation	DB not responding	Error message or fallback mechanism triggered	Pass

CHAPTER 10

PERFORMANCE ANALYSIS

The **performance analysis** evaluates how efficiently and reliably the Transparent Voting System operates under various conditions. It focuses on the system's speed, scalability, resource utilization, and behavior during load and stress scenarios. The goal is to ensure that the system maintains high performance, especially during elections with large numbers of voters.


Performance Analysis
Pre Voting Phase
Pre Voting Phase
Post Voting Phase
User Inputs Aadhar number
Aadhar number is successfully authenticated
OTP authentication is called
Vote is casted
Hash Key is successfully generated
User inputs hash key to successfully submit his vote
Result is displayed

CHAPTER 11

CONCLUSION

The **Transparent Voting System** successfully demonstrates a secure, transparent, and tamper-proof method of conducting elections. By integrating blockchain technology with traditional web development tools, the system overcomes many limitations of conventional voting mechanisms, such as data manipulation, duplicate voting, and lack of transparency.

The project highlights how blockchain's core properties—**immutability, decentralization, and transparency**—can be effectively utilized to build trust in the voting process. From voter registration and authentication to casting votes and result declaration, every stage ensures data integrity and security.

Through thorough testing and performance analysis, the system proved to be stable, efficient, and reliable under simulated conditions. Although the current implementation is best suited for small to medium-scale elections (e.g., college or community elections), it sets a strong foundation for scaling into more robust and real-time national voting platforms.

This project not only fulfills academic objectives but also meaningfully contributes to **Digital Democracy** and **Sustainable Development Goal 16 (Peace, Justice, and Strong Institutions)**. With further enhancements like biometric authentication, real blockchain networks, and large-scale deployment, this system has the potential to revolutionize the electoral process.

CHAPTER 12

REFERENCE

- "Blockchain-Based Electronic Voting System"

  Authors: Ayed, Abir

  Journal: International Journal of Advanced Computer Science and Applications, 2017

  Link: https://thesai.org/Downloads/Volume8No10/Paper_47-Blockchain_Based_Electronic_Voting_System.pdf

- "A survey on blockchain-based electronic voting systems"

  Authors: Tufail, M., Maqbool, A., & Nazir, B.

  Journal: Computers & Electrical Engineering, 2021

  DOI: 10.1016/j.compeleceng.2020.106845

- "An E-Voting System Based on Blockchain and Ring Signature"

  Authors: Chaieb, Chaker, et al.

  Conference: 2019 International Conference on Advanced Systems and Electric Technologies (IC_ASET)