

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590018



Project Synopsis on **BLOCKCHAIN-BASED VOTING SYSTEM**

Submitted in partial fulfilment of the requirement for the award of Bachelor Degree
In

Computer Science and Engineering Submitted By

ADITHYA K P	4HG22CS002
KAVYA R	4HG22CS020
MUSKAN MAURYA	4HG22CS026
ASHWINI PATIL	4HG23CS411

Under the Guidance of
Miss. Zeeba Parveen

B.E
Faculty, Dept of CSE

Head of Dept
Dr. K C Ravishankar
B.E., M.Tech, Ph.D.
Professor and Head Dept of CSE

Under the Guidance of
Dr. Raghu M E
B.E., M.Tech, Ph.D.
Assistant Professor, Dept of CSE

Contents

- 1. Introduction**
- 2. Problem Statement**
- 3. Aim**
- 4. Scope**
- 5. Objectives**
- 6. Requirements**
- 7. Methodology and Design**
- 8. Flowchart**
- 9. Application**
- 10. Implementation**
- 11. Results**
- 12. Work done so far**
- 13. Work to be done**
- 14. Conclusion and References**



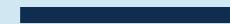
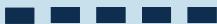
01. Introduction



What it is: A digital platform using blockchain to ensure a **secure, transparent, and tamper-proof** process.

Key Features:

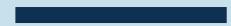
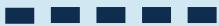
- **Immutable Vote Recording:** Every vote is **permanent and unchangeable**.
- **Real-time Verification:** Builds trust by allowing result verification.
- **Decentralized Consensus:** Eliminates the need for a **single central** authority.
- **Our Project:** A blockchain-based voting system designed specifically for **local elections**.
- **The Goal:** To reduce fraud, ensure fairness, and build complete trust in our local election results.





02. Problem Statement

- **The "Black Box" Problem:** The voting process is opaque, preventing voters from verifying their vote.
- **Centralized Trust:** The system's integrity relies entirely on a single central authority, not a verifiable process.
- **Lack of Public Audits:** Real-time auditing by the public is difficult, leading to potential disputes.
- **Accessibility Barriers:** Physical polling stations and rigid times create hurdles, leading to low voter turnout.



03. Aim



To build a **secure, transparent, and tamper-proof** digital voting system using blockchain, specifically for **local elections**.



The goal is to enhance **community trust** and increase participation by:

- Ensuring voter anonymity and creating immutable records.
- Enabling **real-time verification** for all voters.
- Allowing **secure remote voting** to overcome accessibility barriers.
- Reducing inefficiencies to provide **faster, more reliable election outcomes**.

04. Scope



- **Security & Integrity:** Encrypted, tamper-proof records for all local votes.
- **Transparency & Trust:** A verifiable and auditable process to build **community confidence**.
- **Decentralization:** No central point of control, increasing reliability and removing single points of failure.
- **Anonymity & Privacy:** Guarantees voter confidentiality while ensuring each vote is valid and counted.
- **Accessibility:** Secure remote access to **increase voter turnout** in local elections.
- **Efficiency:** Faster, automated results, lower costs for the municipality, and no manual counting errors.
- **Automation:** Using Smart Contracts for error-free validation and tallying of local ballots.
- **Local Focus:** The system is purpose-built for the specific needs of a local election (e.g., wards, councils, and candidate lists).



05. Objectives



- **To ensure** secure, tamper-proof vote recording for the local election.
- **To implement** a safe and simple authentication process for eligible community voters.
- **To build** a decentralized platform that eliminates central control and enhances public trust.
- **To guarantee** complete voter privacy and the anonymity of their choices.
- **To provide** real-time, transparent results that are publicly verifiable.
- **To design** an accessible and intuitive interface, encouraging higher voter participation in local elections.

06. Requirement Specifications

Hardware:

- Intel i3 / Ryzen 3 or higher
- 4GB RAM (8GB recommended)
- 128GB SSD or more

Software (Tools & Technologies Frontend):

- **Blockchain:** Ethereum
- **Frontend:** HTML, CSS, React.js
- **JavaScript Backend:** Solidity (Smart Contracts), node.js
- **Web3.js Testing:** Ganache
- **Truffle Wallet:** MetaMask



07. Methodology



- **Vote Casting:** Voter submits vote after authentication
- **Validation & Encryption:** Invalid votes rejected; valid ones encrypted
- **Blockchain Storage:** Encrypted votes added immutably to the blockchain
- **Smart Contract Tallying:** Automated, unbiased vote counting
- **Result Evaluation:** Declare winner or trigger runoff if needed
- **Runoff Voting:** Repeat until a clear winner is found



08. Flowchart

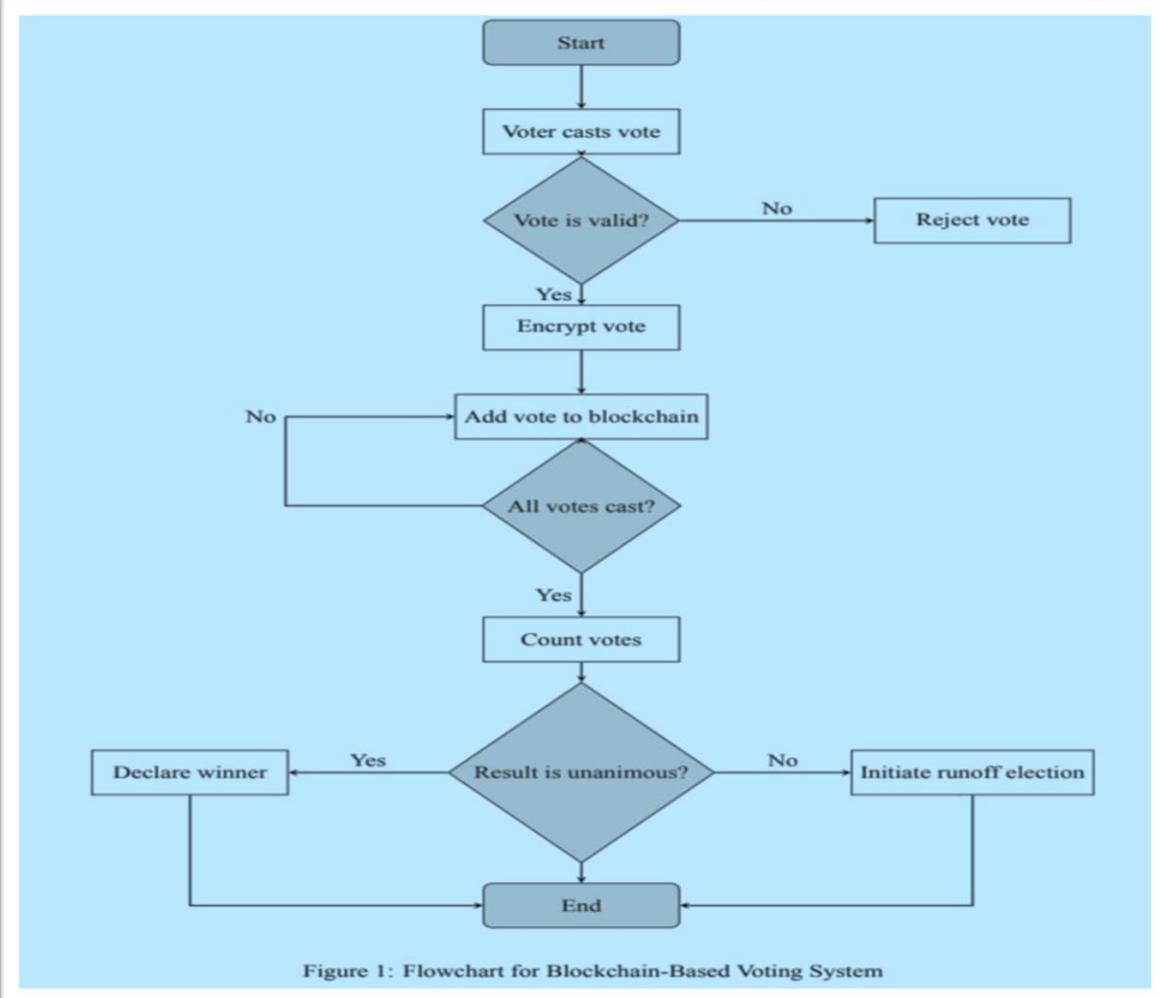


Figure 1: Flowchart for Blockchain-Based Voting System

09. Application



- **Community Voting:**

- Elections for housing societies, neighborhood associations, and resident welfare associations (RWAs).

- **University & Campus Elections:**

- Voting for student councils, unions, clubs, and faculty representatives.

- **Corporate & Organizational Voting:**

- Secure polling for shareholder meetings, board member elections, or internal union votes.

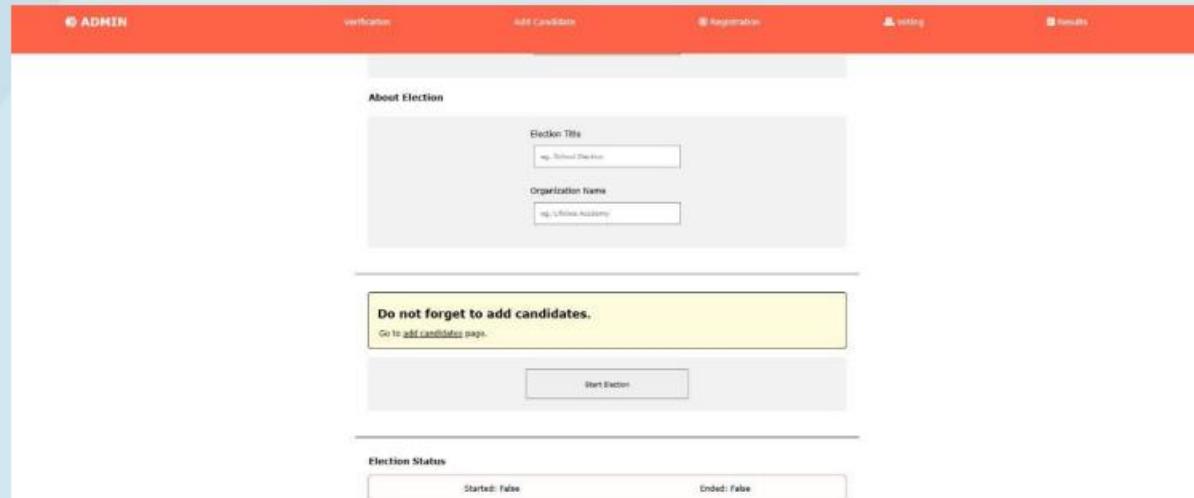
- **Secure Online Polls:**

- Providing a tamper-proof way for any private club, group, or online community to conduct a verifiable poll.



10. Implementation

Figure 6.1: About Admin



"About Admin" section displays the administrator's account address and provides fields to enter their full name and email. It also shows a notification indicating that the election has not yet been initialized, prompting the admin to set it up.

Figure 6.2: User Registration

The screenshot shows a user registration form within a web-based administrative interface. The top navigation bar is orange and contains links for 'ADMIN', 'Verification', 'Add Candidate', 'Registration', 'Voting', and 'Results'. A message in a green box states 'Total registered voters: 1'. The main section is titled 'Registration' and includes a sub-instruction 'Register to vote.' Below this are three input fields: 'Account Address' containing the value '0xD75cd2d86B5cBAA57000DCdcFfCa7B8EfD05a5759', 'Name' containing 'eg. Ava', and 'Phone number *' containing 'eg. 9841234567'. A note below these fields reads: 'Note: Make sure your account address and Phone number are correct. Admin might not approve your account if the provided Phone number nub does not matches the account address registered in admins catalogue.' At the bottom right of the form is a grey 'Update' button.

ADMIN

Verification Add Candidate Registration Voting Results

Total registered voters: 1

Registration

Register to vote.

Account Address

0xD75cd2d86B5cBAA57000DCdcFfCa7B8EfD05a5759

Name

eg. Ava

Phone number *

eg. 9841234567

Note:

Make sure your account address and Phone number are correct.
Admin might not approve your account if the provided Phone number nub does not matches the account address registered in admins catalogue.

Update

User register with account addresss,name,Phone number

Figure 6.3: Home Page

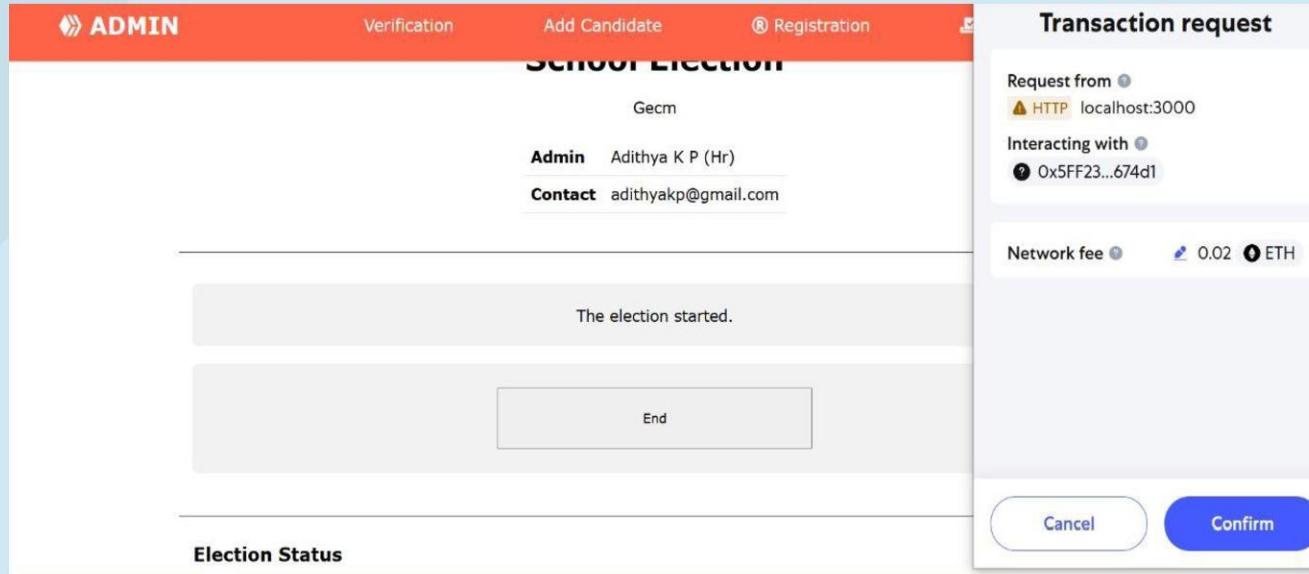
The screenshot shows a web application interface for a school election. At the top, there is a red navigation bar with the following items: a logo labeled "ADMIN", "Verification", "Add Candidate", "Registration", and a user icon. Below the navigation bar, the title "SCHOOL ELECTION" is displayed in large, bold, black letters. Underneath the title, the acronym "Gecm" is shown. Below "Gecm", the roles "Admin" and "Adithya K P (Hr)" are listed, along with their contact information: "Contact" and "adithyakp@gmail.com". A horizontal line separates this section from the main content area. In the main content area, there is a message "The election started." followed by a button labeled "End". Another horizontal line separates this from the "Election Status" section, which contains the text "Election Status". To the right of the main content area, a modal window titled "Transaction request" is open. It displays the following details:

- Request from: **HTTP** localhost:3000
- Interacting with: **0x5FF23...674d1**
- Network fee: 0.02 ETH

At the bottom of the modal, there are two buttons: "Cancel" and "Confirm".

It represents Home page

Figure 6.4: Election End



The **"Election End"** section allows the admin to officially end the election process. Upon clicking the "End" button, a blockchain transaction is initiated, requiring confirmation along with a network fee to finalize and record the election's conclusion.

11.Results

Figure: Election Rules & Guidelines

The screenshot shows a dark-themed web interface for election management. At the top, there is a navigation bar with tabs: Admin (which is active), Verification, Add Candidate, Registration, Voting, Results, and a Light mode switch. Below the navigation bar, the main content area has a blue header titled "Election Rules & Guidelines". The page is divided into four sections: "Voter Registration", "Voting Process", "Security & Transparency", and "Election Timeline".

- Voter Registration:**
 - Each voter must register with valid name and phone number
 - Only one registration per blockchain address
 - Admin must verify each registration before voting
 - Registration is required before the election starts
- Voting Process:**
 - One vote per verified voter
 - Votes are recorded on the blockchain
 - Voting is anonymous and secure
 - Cannot change vote once submitted
- Security & Transparency:**
 - All transactions are recorded on blockchain
 - Results are tamper-proof and verifiable
 - Admin cannot manipulate votes
 - Full transparency in the voting process
- Election Timeline:**
 - Registration period: Before election starts
 - Voting period: After admin starts election
 - Results: Available after election ends
 - No voting after election is closed

"Election Rules & Guidelines" This page explains the rules of voter registration, secure voting process, and blockchain-based transparency. It highlights one-vote-per-user policy, tamper-proof results, and clear election timelines to ensure fair and trustworthy elections."

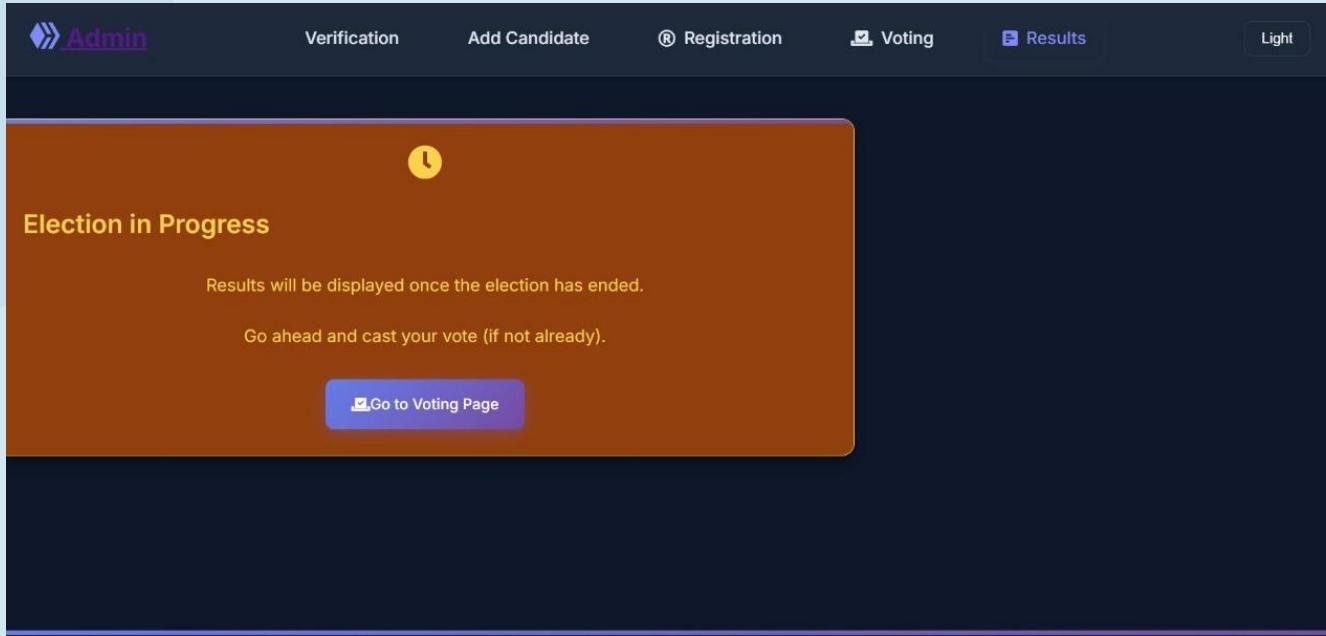
Figure : “Election Results”

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links: "Admin" (highlighted in blue), "Verification", "Add Candidate", "Registration", "Voting", "Results" (highlighted in blue), and a "Light" mode switch. Below the navigation bar, the title "Election Results" is displayed with a chart icon. A message "Total candidates: 4" is shown above a table. The table has three columns: "ID", "CANDIDATE", and "VOTES". The data is as follows:

ID	CANDIDATE	VOTES
0	adithya	1
1	kavya	0
2	muskhan	0
3	ashwini	0

“Election Results” This page displays the final vote count for each candidate in a transparent manner. Results are recorded on the blockchain, ensuring accuracy, security, and tamper-proof verification.

Figure : “Election in Progress”



“Election in Progress” This page indicates that the election is currently active and results will only be visible after completion. It allows verified voters to cast their votes during the voting period.”

Figure : Voter Verification

The screenshot shows a web-based voter verification interface. At the top, there is a navigation bar with links for Admin, Verification, Add Candidate, Registration, Voting, Results, and Light mode. Below the navigation bar, a blue header bar displays the text "List of registered voters". The main content area contains a table with the following data:

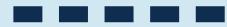
ACCOUNT ADDRESS	0x8d53EfAa4F0a98c8f7Ebb64fb77390aFe10d4c46
NAME	adithya p
PHONE	6565656565
VOTED	False
VERIFIED	False
REGISTERED	True

At the bottom of the table, there is a white button labeled "Approve".

“Voter Verification” This page allows the admin to verify registered voters before participation. It displays voter details like account address, name, phone, and status, ensuring only verified users can cast their votes.”

12. Work Done So Far:

- Add candidate functionality (managed by Admin)
- Implement voter registration (by Admin)
- Enable voter verification process
- Develop voting interface for users
- Create result interface for displaying outcomes
- Improve overall UI/UX design for better user experience



13. Conclusion

- Our blockchain-based system provides a **secure, transparent, and tamper-proof** solution perfectly tailored for **local-level elections**.
- By leveraging decentralization, we solve key challenges of **trust and accessibility** for community members.
- This technology ensures voter privacy and provides **verifiable, trustworthy results**, giving every member confidence in the outcome.
- This model can revolutionize democratic processes not just in local elections, but in any local-scale vote, from **university councils to housing societies**, making them more efficient, inclusive, and credible.





14. References

- Nakamoto, S. – *Bitcoin Whitepaper*
- Ethereum Foundation – ethereum.org
- Hyperledger – hyperledger.org
- Swan, M. – *Blockchain: Blueprint for a New Economy*
- IBM Blockchain – ibm.com/blockchain
- IEEE & Springer articles on blockchain voting



Thanks!

