

Decentralised blockchain voting/election system:

First of all, why are we building a decentralised voting/election system, why is need of this and how it is better than a conventional system:

- **Transparency:** Transactions on a blockchain are transparent and immutable, meaning that once a vote is recorded, it cannot be altered or tampered with.
- **Security:** Blockchain technology utilizes cryptographic algorithms to secure transactions, making it extremely difficult for unauthorized parties to manipulate or hack the voting process. Decentralization also makes it harder for any single entity to control or manipulate the outcome of an election.
- **Decentralization:** Unlike centralized online voting systems, which are vulnerable to single points of failure and potential manipulation by a central authority, decentralized blockchain voting systems distribute control and decision-making across a network of nodes. This reduces the risk of fraud and ensures that no single entity can control the outcome of an election.
- **Better Accessibility**(if we compare it with traditional physical polling booths): With internet access, voters can participate from anywhere in the world, eliminating the need for physical polling stations.
- **Reduced Costs** (compared to physical polling booths): By eliminating the need for physical ballot printing, distribution, and manual vote counting, blockchain voting systems can potentially reduce the costs associated with conducting elections. This could make it easier for governments to hold more frequent or specialized elections.

Of all the benefits of a decentralised blockchain voting system, the most important and main are transparency, security and decentralization

Uniqueness and newness compared to the existing decentralised voting system:

We are Offering different types of voting in a decentralized blockchain system, it is like giving organizers a toolbox with various tools instead of just one. They can pick the tool that fits their needs best. For example, sometimes it's not about who gets the most votes overall, but about how people rank their choices. This can make sure more people feel represented. By having this variety, the system becomes more flexible, transparent, and educational. It helps elections better reflect what people really want, especially in more complex situations. Overall, it's about giving more power and options to the people organizing the elections.

In our project, you're not just sticking to the usual "most votes wins" approach. Instead, you're offering a whole menu of voting options for administrators to choose from. Here's a glimpse into some of the exciting voting methods you're providing:

1. **Plurality Voting:** This is the classic "most votes wins" method. Voters choose their favorite candidate, and the one with the most votes wins. It's simple and straightforward, but it doesn't always reflect the full spectrum of voter preferences.
2. **Ranked Choice Voting:** With this method, voters rank candidates in order of preference. If no candidate gets a majority of first-choice votes, the candidate with the fewest votes is

eliminated, and their votes are redistributed based on voters' second choices. This process continues until a candidate has a majority. It's great for ensuring that the winner has broad support among voters.

3. Approval Voting: Instead of picking just one candidate, voters can choose to approve as many candidates as they like. The candidate with the most approvals wins. It's a simple way for voters to express support for multiple candidates without having to rank them.
4. Borda Count: In this method, voters rank candidates, and each ranking earns points. The candidate with the most points overall wins. It's a way to take into account not just voters' top choices, but also their preferences among all the candidates.
5. Cumulative Voting: Here, voters are given a certain number of votes, and they can allocate them however they choose among the candidates. They can give all their votes to one candidate or spread them out among several. It's a way for voters to express varying degrees of support for different candidates.
6. Range Voting: In range voting, voters assign a score or rating to each candidate, typically within a specified range (e.g., 0 to 10). The candidate with the highest average score across all voters wins the election.
7. Condorcet Method: In the Condorcet method, voters rank candidates in order of preference. The winner is the candidate who would beat every other candidate in a head-to-head matchup according to the ranked preferences of the voters.
8. Single Transferable Vote (STV): Single Transferable Vote is a preferential voting system used in multi-seat elections. Voters rank candidates in order of preference, and seats are allocated based on a specific formula that takes into account both voters' first preferences and the transfer of surplus votes from elected candidates or the redistribution of votes from eliminated candidates.
9. Instant Runoff Voting (IRV): Instant Runoff Voting, also known as ranked choice voting, allows voters to rank candidates in order of preference. If no candidate receives a majority of first-choice votes, the candidate with the fewest votes is eliminated, and their votes are redistributed to the remaining candidates based on voters' second choices. This process continues until a candidate receives a majority of votes.

Feature of voting system application:

- Home page
- Title: Decentralized Voting Platform
- Logo
- Login to Voter
 - Username input field
 - Password input field
 - "Login" button
 - "Forgot Password?" link

- Login to Admin
 - Username input field
 - Password input field
 - "Login" button
 - "Forgot Password?" link
- Features after login to Admin:
 - Create types of election
 - Max no. of people allowed to vote
 - Date time of election
 - Manage voters,
 - Manage candidates,
 - Check the status of votes,
 - Declaring results
 - E.t.c
- Voter/user:
 - Title: Welcome to the Voting Platform
 - Components:
 - Logo
 - Input field for voter ID or unique identifier
 - "Verify Identity" button
 - Select election,
 - Verify ID
 - Vote,
 - See result

The software we'll be using for the project

- IDE
 - Visual studio code
 - Remix IDE
- Metamask (connect to blockchain addresses to connect to solidity smart contract)
- Ganache
- Solidity (write smart contract in solidity and upload it into any blockchain network)
- React JS (application)
- Ethers.JS
- Hardhat (application will run in react JS and will communicate to blockchain through hardhat ether.js)
- any browser