BLOCKCHAIN VOTING

DOMAIN: NETWORKING

405 FOUND

TEAM:

PRANAV HEMANTH: PES1UG23CS433

SAMPRITI SAHA: PES1UG23CS505

KSHITIJ KOUSHIK KOTA: PES2UG23CS291

PRANAVJEET NAIDU: PES1UG23CS586

TABLE OF CONTENT

PROBLEM STATEMENT

Improving the election voting system using blockchain to make voting more:

- accessible
- transparent
- efficient

SOLUTION

- Accessibility: Everyone with a phone or computer can cast their vote irrespective of their location
- Transparent: An open-source blockchain implements the system to increase trust.
- Efficient: Elections can be held faster, for cheaper whilst increasing integrity in the process.

ABOUT US

- An Open-Source blockchain platform for citizens to cast votes seamlessly, anonymously, and securely
- Modernize voting by using blockchain, enabling people to vote from anywhere with their devices
- Ensures transparency: blockchain makes tampering nearly impossible
- Streamlines the process, reducing time and making elections more efficient
- Votes remain anonymous, protecting voter privacy

TECH STACK

BLOCKCHAIN PLATFORM	ETHEREUM FABRIC SMART CONTRACT
DEVELOPMENT TOOL	SOLIDITY
FRONTEND DEVELOPMENT	REACT.JS
BACKEND DEVELOPMENT	NODE.JS + EXPRESS.JS
DATABASE	MONGODB
API INTEGRATION	REST-APIS
DEVELOPMENT ENVIRONMENTS	REMIX IDE, VS CODE

SMART CONTRACTS

Written using Solidity - later to be deployed on Ethereum
This includes the following functionalities:

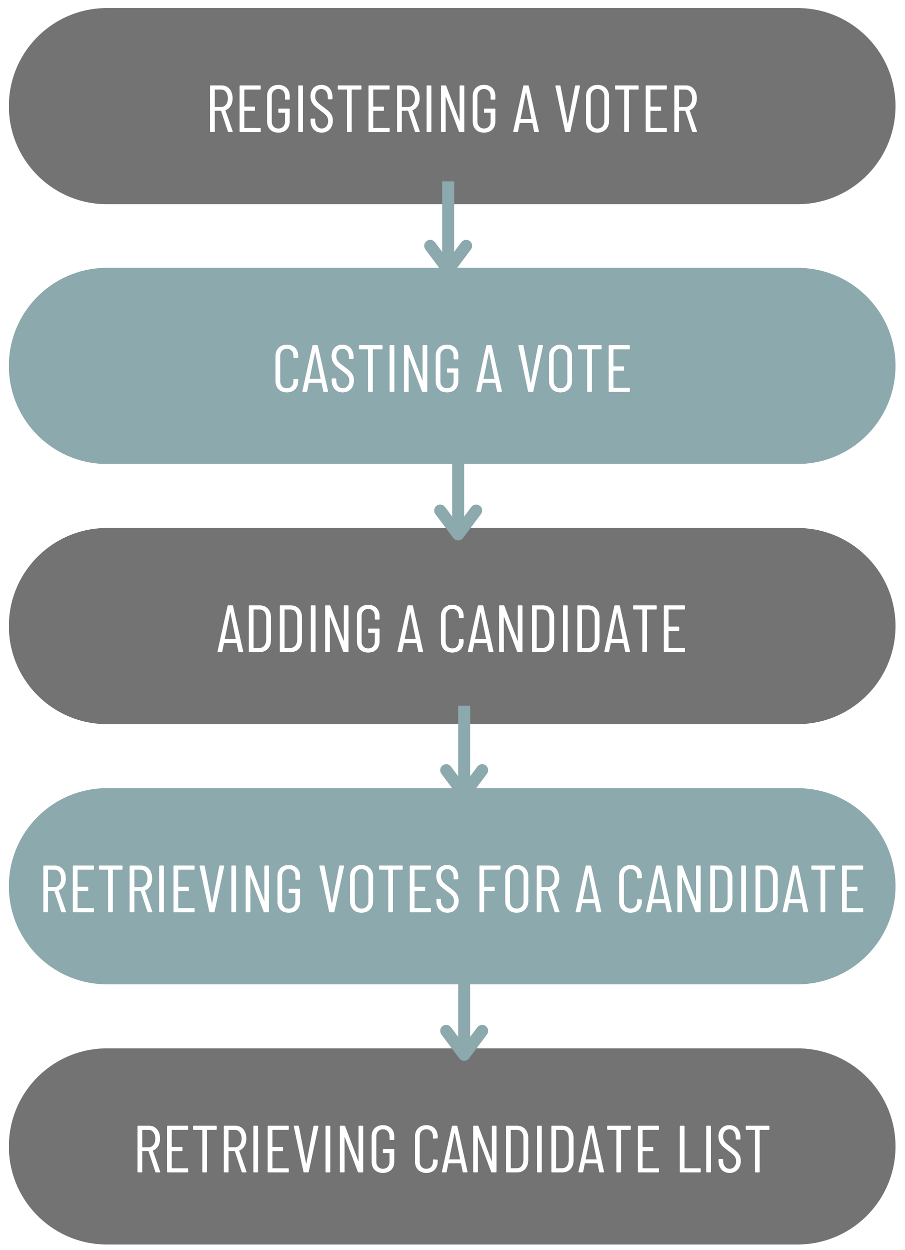
voter to register: marking as registered (voters mapping.)

registered voter: cast a vote for a candidate, increment vote count (<votesReceived> mapping), marks the voter as voted

contract owner to add a candidate to the list of candidates

allows election commision to retrieve the total votes received by a candidate

allows anyone to retrieve the list of candidates.



DEPLOYMENT ON ETHEREUM

1. Remix IDE: Environment to deploy smart contracts
2. Remix VM: to deploy and test smart contracts in a simulated blockchain environment
3. Deployment Process:
    - Write Solidity smart contract in the Remix IDE editor.
    - Compile the smart contract.
    - Deploy the smart contract selecting JavaScript VM as the environment.
4. Interacting with Smart Contracts:
    - You can call functions, send transactions, and view transaction details.

BACKEND NODE.JS EXPRESS.JS

Stores data, validates transactions, executes smart contracts for the voting process	BLOCKCHAIN NODE
Solidity: handle voter registration, ballot creation, vote casting, and result tallying	SMART CONTRACTS
RESTful interface for client interaction with the blockchain and smart contracts	API LAYER
Stores non-sensitive data like voter registration details, ballot configurations, and election results	DATABASE
Ensures only eligible voters can cast votes, integrating with identity verification services.	AUTHENTICATION AND AUTHORIZATION
Implements encryption, secure API endpoints, and transaction logging for security and auditing.	SECURITY AND AUDITING
Handle large numbers of users and transactions, especially during peak voting periods.	SCALABILITY AND PERFORMANCE

FRONTEND USING REACT.JS

ADMIN PAGE

- REGISTER VOTERS
- ADD CANDIDATES
- RETRIEVE ELECTION RESULTS
- SHOW CANDIDATE LIST

VOTER PAGE

- AUTHENTICATE VOTER
- VOTE FOR CANDIDATE

FRONTEND USING REACT.JS

1. Admin Page:

- Register Voters: Form for election commission to register voters by providing voter details.
- Add Candidates: Form to add candidates to the election by entering candidate details.
- Retrieve Election Results: Display of election results including candidate names and vote counts.
- Show Candidate List: Display of the list of candidates running in the election.

2. Voter Page:

- Authenticate Voter: Form for registered voters to authenticate themselves (e.g., using a voter ID or other credentials).
- Vote for Candidate: Display of the list of candidates with an option to select and cast a vote for a favored candidate.

Blockchain Voting

Business Model Canvas

KEY PARTNERS

- Development Team: Developers and experts in blockchain technology, security, and voting systems.
- External Services: Identity verification services, cloud providers for hosting, and other third-party services.

KEY ACTIVITIES

- Developing Smart Contracts
- Integrating with Frontend and Backend:
- Ensuring Security

KEY RESOURCES

- Blockchain Infrastructure
- Technical Expertise
- Data Storage

VALUE PROPOSITIONS

- Transparent and Secure Voting: Utilizing blockchain technology to ensure transparency, security, and integrity of the voting process.
- Convenience and Accessibility: Allowing voters to cast their votes online from anywhere, increasing voter turnout.

CUSTOMER RELATIONSHIPS

- User Support: to voters and administrators
- Improvement: Collecting feedback from users

CHANNELS

- Online Platform: interface for voters
- Communication Channels: Using email, social media, etc. to inform voters about the voting process

CUSTOMER SEGMENTS

- Voters: Individuals eligible to vote in elections
- Election Administrators: Organizations or individuals responsible for managing and overseeing the election process.

COST STRUCTURE

- Development Costs: Expenses related to developing and deploying the voting system.
- Maintenance and Support: Costs associated with maintaining the platform and providing user support.
- Security Measures: Investments in security to protect the integrity of the voting process.

REVENUE STREAMS

- Government Funding: Includes overheads such as charging a small fee for each transaction (e.g., voter registration, vote casting) to cover the operational costs of the platform.
- Subscription Model: Offering a subscription-based model for organizations conducting elections regularly.

THANK YOU