Université Abdelmalek Essaâdi

Faculté des Sciences de Tétouan

Filière : Sciences Mathématiques et Informatiques

Final Study Report:

DECENTRALIZED VOTING APPLICATION BASED ON ETHEREUM BLOCKCHAIN

Presented by:
SIAKA Riad
SODOR Anass

Date of the defense:
13 JULY 2023

JURY :

Full Name	Establishment	Quality
Pr. M.L. BEN MAATI	Faculty of Sciences, Tetouan	President
Pr. ATTARUIS Hicham	Faculty of Sciences, Tetouan	Examiner
Pr. YOUNES Ali	Faculty of Sciences, Tetouan	Academic Supervisor

dedications

“At the beginning of this work, we thank God for granting us the courage, the determination, and the patience to successfully complete this modest work.

Second, we would like to dedicate this graduation project to our beloved parents, whose unwavering love, boundless support, and constant belief and encouragement have been invaluable to us.

Furthermore, we would like to express our deep gratitude to our supervisor, Mr. ALI YOUNESS, who honored us by overseeing this work and providing us with valuable advice and guidance.

We also extend our thanks to the members of the jury for accepting to examine and evaluate our work. We are also grateful to the teachers who contributed to our progress over the past 3 years.

Finally, we would like to thank all the individuals who, whether near or far, supported us throughout our journey.

Thank you.”

abstract

Blockchain technology is a distributed ledger technology that offers security, transparency, and trust by storing information in a decentralized way. It is being used in various industries, including finance, supply chain management, and healthcare.

A Decentralized Voting Application is a promising idea that improves transparency, streamlines processes, and reduces fraud risks. This can lead to increased trust, easier access for voters, secure data, and lower costs.

A smart contract automates tasks and reduces errors, ensuring compliance with election rules. Privacy is protected through encryption.

While challenges exist, the use of blockchain in this DApp has the potential to transform democracy, fostering trust, transparency, and inclusivity.

Keywords: Blockchain; Decentralized application; Ethereum;

Smart contract;

contents

list of figures

tables

introduction

Introducing a groundbreaking Decentralized voting system, an innovative and transparent solution designed to empower individuals in decision-making processes. By leveraging blockchain technology, this decentralized application (DApp) ensures the utmost security, transparency, and efficiency in voting.

Replacing traditional paper-based methods, our DApp offers a digital platform accessible to anyone with an internet connection. Through the utilization of smart contracts, the system guarantees the integrity of the voting process, effectively eliminating any possibility of tampering or fraud.

Participating in the voting process is both straightforward and secure. Voters can easily create an account and verify their identity, enabling them to contribute to the decision-making process. Each vote is recorded immutably on the blockchain, ensuring that the results cannot be altered or manipulated. This high level of transparency builds trust among participants, guaranteeing that every vote holds significance.

The DApp 's user-friendly interface simplifies the voting experience, allowing individuals to securely and conveniently cast their votes. Real-time updates are facilitated, providing continuous visibility of the current vote count and overall progress.

Moreover, its’s enables efficient and cost-effective management of elections by eliminating the need for physical infrastructure and manual vote counting. The automated process saves time and resources for both voters and administrators alike.

In summary, our Decentralized voting application harnesses the power of blockchain technology to establish a secure, transparent, and user-friendly platform for democratic decision-making. Through this revolutionary approach to the voting process, we aim to enhance trust, participation, and accountability in all types of elections and decision-making processes.

Chapter 1: Project Context

I. Blockchain Technology

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system.

A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT).

Blockchain is a type of DLT in which transactions are recorded with an immutable cryptographic signature called a hash.

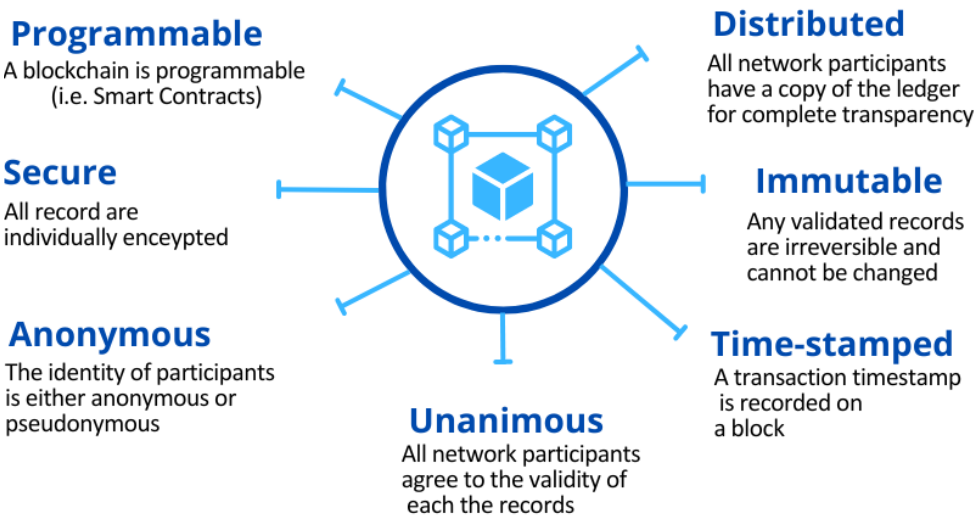

Programmable
A blockchain is programmable (i.e. Smart Contracts)
Secure
All record are individually enceypted
Anonymous
The identity of participants is either anonymous or pseudonymous
Unanimous
All network participants agree to the validity of each the records
Distributed
All network participants have a copy of the ledger for complete transparency
Immutable
Any validated records are irreversible and cannot be changed
Time-stamped
A transaction timestamp is recorded on a block

Figure 1 : Properties of Distributed Ledger Technology (DLT)

1. How blockchain data is stored and secured

Blockchain works by including the identifier of the last block into the identifier in the following block to create an unbreakable and immutable chain. But as more and more blocks are added, how does the data remain manageable?

The key to keeping blockchain data manageable – and secure – is through an algorithm called hashing in combination with a consolidating data structure known as a Merkle Tree.

1.1 Blocks

The different recorded transactions are grouped into blocks. After recording the recent transactions, a new block is generated, and all the transactions will be validated by the miners, who will analyze the complete history of the blockchain. If the block is valid, it is timestamped and added to the blockchain. The transactions it contains are then visible throughout the network. Once added to the chain, a block cannot be modified or deleted, ensuring the authenticity and security of the network.

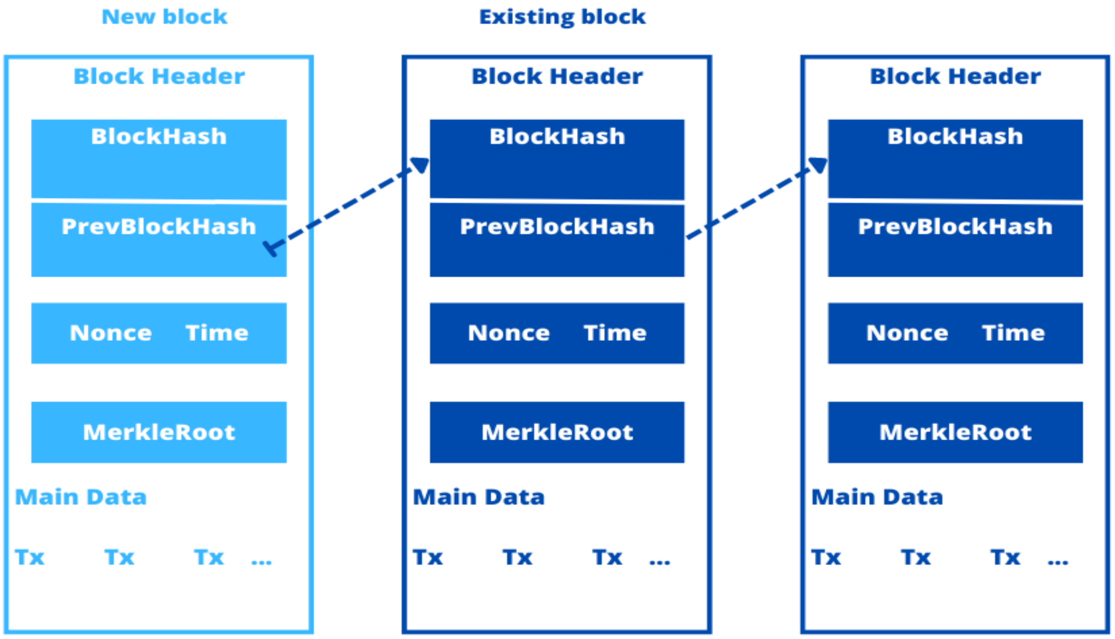


Figure 2 : Chain of blocks

Each block in the chain consists of the following elements:
Transactions performed on the network.
A hash checksum used as an identifier.
The checksum of the previous block (except for the first block in the chain, called the genesis block).
A measure of the amount of work that was required to produce the block. This is defined by the consensus method used within the chain, such as the nonce "proof of work."

1.2 What is hashing

When a transaction has been verified and needs to be added to a block in a chain, it will be put through a hash algorithm to convert it into set of unique numbers and letters, similar to what would be created by a random password generator. Then two transaction hashes will be combined, and put through the hash algorithm to produce another unique hash. This process of combining multiple transactions into new hashes continues until finally there remains just one hash – the 'root' hash of several transactions.

What makes hashes unique, and a key security feature for blockchains, is that they only work one way. While the same data will always produce the same hash of numbers and letters, it is impossible to 'un-hash', or reverse the process, using the numbers and letters to decipher the original data.

1.3 What is Merkle Tree?

If the hashing process is repeated with exactly the same transactions, exactly the same hashes will be created. This allows anyone using the blockchain to check that the data has not been tampered with, because any change in any part of the data will result in a completely different hash, affecting every iteration of hashes all the way to the root. This is known as a Merkle Tree.

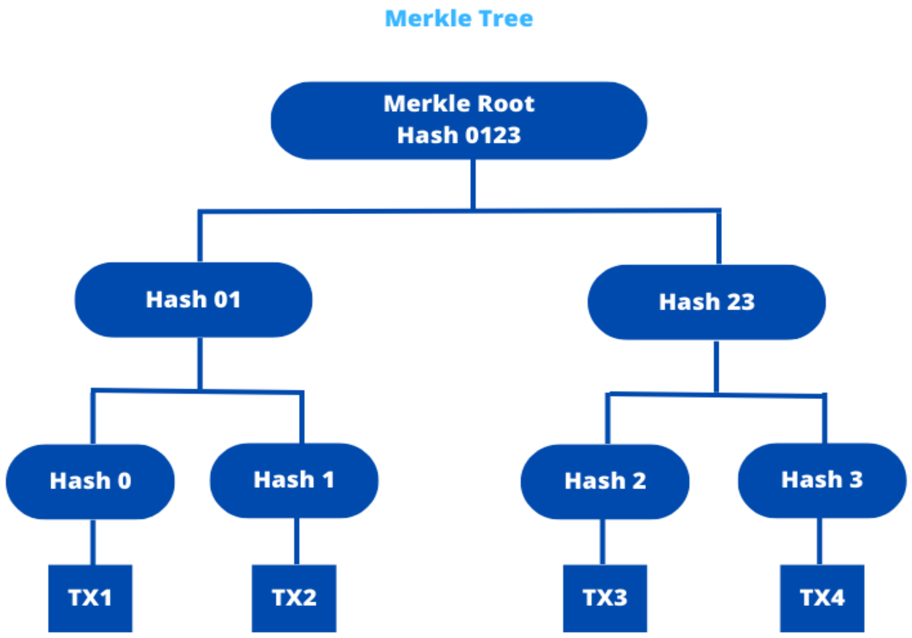


Figure 3 : Merkle Tree

2. Distributed Ledger

Distributed Ledger Technology (DLT) is a decentralized and transparent system for recording and verifying transactions or information across multiple nodes or participants. It offers a secure and tamper-resistant method of maintaining a shared database, eliminating the need for a central authority or intermediary.

2.1 Decentralization: Redefining Trust

DLT enables decentralization by distributing copies of the ledger across multiple participants, eliminating the reliance on a single central authority. This decentralized approach enhances trust, as no single entity has exclusive control over the data, ensuring transparency and reducing the risk of fraud or manipulation.

2.2 Immutable and Tamper-Resistant: Preserving Data Integrity

Once a transaction is recorded on the distributed ledger, it becomes nearly impossible to alter or tamper with the data. Each new transaction is cryptographically linked to previous transactions, forming a chain of blocks, where each block contains a unique identifier called a hash. Any attempt to modify a transaction would require the consensus of the majority of participants, making it highly secure and resistant to fraudulent activities.

2.3 Consensus Mechanisms: Achieving Agreement

Consensus mechanisms play a crucial role in distributed ledger systems. They ensure that all participants agree on the state of the ledger and validate transactions. Various consensus algorithms, such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), are employed to achieve agreement among participants while maintaining security and efficiency.

3.Blockchain Benefits

Decentralized: Because blockchains are managed by a network of nodes rather than a central authority, they are fully decentralized. This prevents any one entity from having any control over the network.

Transparent: Transactions on a blockchain are constantly being recorded and stored on the blockchain across nodes. This means that all participants can view all transactions on the network in real-time.

Immutable: Blockchains are designed to enable permanent record keeping so that stored data cannot be altered after being added. This makes it an extremely stable and reliable record-keeping system.
Secure: It is hard to change or destroy blockchains because of its distributed nature. For example, if someone hacked into one of the computers on the network and altered information there, the network would remain unaffected.

II. Ethereum Blockchain

1.Ethereum Blockchain

Ethereum is a decentralized, open-source blockchain platform that enables the development and execution of smart contracts and decentralized applications (DApps). It was proposed in late 2013 and launched in 2015 by a programmer named Vitalik Buterin.

2.Ethereum virtual machine

The EVM (Ethereum Virtual Machine) is the execution environment for the bytecode of Ethereum smart contracts. Each node in the network runs the EVM. All nodes execute all transactions that target smart contracts using the EVM, so every node performs the same computations and stores the same values. Transactions that only transfer Ether also require computation, meaning to check if the address has a balance or not and deduct the balance accordingly

Each node executes the transactions and stores the final state for various reasons. For example, if there is a smart contract that stores the names and details of all participants in a party, whenever a new person is added, a new transaction is broadcasted on the network. To have any node in the network display the details of all people participating in the party, they simply need to read the final state of the contract .

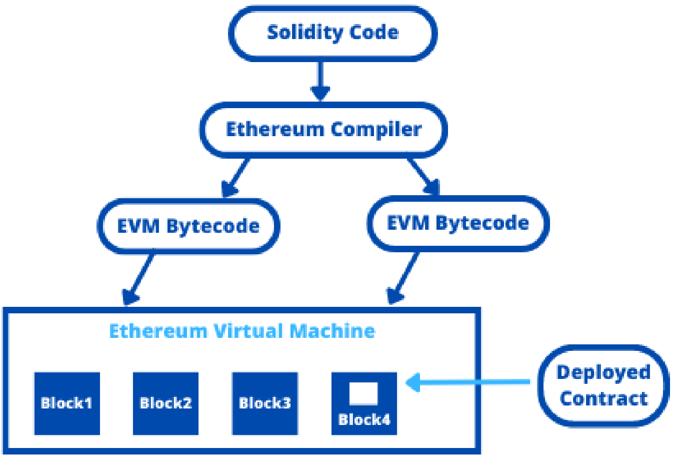


Figure 4 : Ethereum Virtual Machine

3. Consensus Mechanisms

3.1 Proof of work (PoW)

The first blockchain implementation, Bitcoin, uses a method called proof of work to validate transactions. In this method, miners (nodes that validate transactions) compete to solve a complex mathematical problem. The first miner to solve the problem gets to add the next block of transactions to the blockchain and is rewarded with bitcoins.

The proof of work algorithm ensures that only valid transactions are added to the blockchain. This is because it is very difficult to solve the mathematical problem unless you have the correct data. If a miner tries to add an invalid transaction to the blockchain, their block will not be accepted by the rest of the network.

The blockchain also uses a rule called the longest chain to decide which block of transactions is the most recent and valid. This means that if two miners solve the mathematical problem at the same time, the block that is added to the longest chain will be considered the valid block.

This process of validating transactions and adding blocks to the blockchain is what makes Bitcoin and other blockchains secure and tamper-proof.

Here are some additional details about the proof of work algorithm:

- The mathematical problem that miners need to solve is called a nonce. The nonce is a random number that is added to the block of transactions. The miner then needs to hash the block, which means running it through a mathematical function that produces a unique string of numbers and letters.
- The goal of the miner is to find a nonce that produces a hash that is less than or equal to a certain target value. The target value is constantly changing, so it becomes more and more difficult to find a nonce that meets the target value as more blocks are added to the blockchain.
- The miner that finds the nonce first gets to add the block of transactions to the blockchain and is rewarded with bitcoins.
- The proof of work algorithm is very energy-intensive, as miners need to use powerful computers to solve the mathematical problems. However, it is also very secure, as it is very difficult for anyone to cheat the system.

3.2 Proof of Stake (PoS)

Proof-of-stake (PoS) is a consensus mechanism used by some blockchains to validate transactions and add new blocks to the blockchain. In PoS, miners are chosen based on the amount of cryptocurrency they have staked, or locked up, in the network. The more cryptocurrency a miner stakes, the more likely they are to be chosen to validate the next block.

When Ethereum transitioned to PoS in 2022, it required miners to stake at least 32 ETH in order to participate in block validation.

However, miners who do not have 32 ETH can still participate in PoS by joining a stake pool. Stake pools are groups of miners who pool their resources together to increase their chances of being chosen to validate blocks.

When a miner successfully validates a block, they are rewarded with a block reward, which is a combination of newly minted cryptocurrency and transaction fees. The amount of block reward is determined by the specific PoS blockchain. For example, the block reward for Ethereum is currently 2 ETH.

If a miner engages in malicious activity, such as double-spending or attempting to attack the network, they will lose the cryptocurrency they have staked. This provides a strong incentive for miners to act honestly, as they risk losing their investment if they are caught cheating.

It is important to note that in order to gain control of the network, a malicious actor would need to control more than 50% of the stake in the network. This is considered to be very unlikely, as it would require a significant amount of cryptocurrency and would be very expensive to do.

Overall, PoS is a more energy-efficient and secure consensus mechanism than proof-of-work (PoW), which is the consensus mechanism used by Bitcoin. PoS is also more scalable than PoW, as it does not require miners to compete with each other to solve complex mathematical problems. As a result, PoS is becoming increasingly popular as a consensus mechanism for blockchain networks.

4.Data Integrity:

Data integrity is a fundamental concept in blockchains, including the Ethereum blockchain. Simply put, data integrity refers to the trustworthiness and reliability of the information stored in the blockchain.

In Ethereum, data integrity is ensured through the use of cryptographic hashes and decentralized consensus. Here's how it works:

- Cryptographic hashes: Ethereum employs a cryptographic hashing function called Keccak-256 (also known as SHA-3) to convert data into a unique digital fingerprint, known as a hash. Regardless of the input data size, the resulting hash will always have a fixed size of 256 bits. This allows for the creation of a unique fingerprint for each set of data, whether it's a transaction, a block, or any other item stored on the Ethereum blockchain.
- Block structure: Ethereum stores data in a block structure. Each block contains a header and a set of transactions. The block header includes metadata, including the hash of the previous block, thus forming a blockchain. Each block also contains a hash of the set of transactions it includes.
- Decentralized consensus: Ethereum utilizes a consensus algorithm called Proof of Stake (PoS) since the implementation of the Ethereum 2.0 upgrade. In this system, validators hold and lock a certain amount of Ether (ETH) to participate in the transaction validation process. They are responsible for verifying transactions, grouping them into blocks, and validating their integrity by performing calculations based on the hash of the previous block. Consensus is achieved when the majority of validators agree on the validity of a block, and it is highly challenging for an attacker to manipulate data without triggering alerts in the network.
- Data validation: Each node participating in the Ethereum network verifies the integrity of data by recalculating the hashes of blocks and comparing them with the hashes stored

on the blockchain. If the hashes match, it means the data remains unchanged, and integrity is preserved. If a node detects any inconsistency or attempt to modify the data, it can reject fraudulent blocks and continue working on the branch of the blockchain that adheres to the consensus rules.

By combining these elements, Ethereum ensures data integrity by providing a robust mechanism that guarantees the immutability and protection against manipulation of the data stored on the blockchain. This enables users to trust the stored information and build reliable decentralized applications on the Ethereum platform.

III. Smart Contract

1.Smart contract fundamentals

Smart Contract is a program that runs inside the Ethereum Blockchain executed by Ethereum Virtual Machine. Smart Contract is an immutable program, meaning once the code is written and deployed to a blockchain, it cannot be updated or rewritten. Vitalik describes this concept as follows: “code is law”. The language used in Ethereum Smart Contract is Solidity language. After the code is written and ready to be deployed, the developers have the option to deploy it to Mainnet, which is the real network, and it uses real Ether. If the developers want to test their Smart Contract, they can deploy their Smart Contract to four of the Ethereum Testnets: Ropsten, Kovan, Rinkeby, and Goerli. These Testnets do not use real Ether. Instead, the developers can ask for Ether from one of these Testnets’ faucets.

A Smart Contract consists of state variables, events, modifiers, and functions. Each function call that mutates the state variables inside the Smart Contract will be a transaction, and each transaction will cost a certain amount of “gas”. The amount of gas spent will depend on the complexity and the memory of the function. Other functions such as return function or pure function do not consume gas as long

as they are not called from another mutative function and the return function does not mutate the values of the state variables.
The biggest advantage when using Smart Contract is that there is practically no downtime since the blockchain is maintained by millions of users. As long as the Ethereum blockchain network is still up, the Smart Contract will still be valid.

Figure 5 : Smart Contract

IV. Decentralized application (DApp)

1.DApp Definition

A DApp, short for decentralized application, is an application that operates on a decentralized network, typically a blockchain. Unlike traditional applications, which are usually centralized and rely on a single authority or organization to control and manage the application and its data, DApps are designed to be decentralized, transparent, and often open source.

DApps leverage the underlying blockchain technology to provide various functionalities and features. They typically use smart contracts, which are self-executing contracts with the terms of the agreement directly written into code. Smart contracts allow for automated and trustless execution of transactions and agreements, eliminating the need for intermediaries.

2.Strenght points of DApp

There are many advantages to creating a DApp that a typical centralized architecture cannot provide:
Decentralization: DApps are not controlled by any central authority. They operate on a peer-to-peer network of computers, typically a blockchain, where all participants have equal control and influence over the application.

Transparency: The rules and logic of a DApp are typically open and visible to all participants. This transparency ensures that all users can verify the operations and transactions performed within the application.

Security: DApps leverage the security features of the underlying blockchain technology, such as cryptographic encryption and consensus mechanisms, to provide a high level of security and immutability.

Tokenization: Many DApps have their own native tokens or cryptocurrencies that are used to facilitate transactions, incentivize users, or govern the operation of the application.

DApps can be developed for various purposes, such as decentralized finance (DeFi), supply chain management, gaming, social networking, and more. They aim to provide users with increased control over their data, reduced reliance on intermediaries, and the potential for new economic models and opportunities.

V. Presentation

1.Problem Statement

Voting must adhere to five essential characteristics: transparency, uniqueness, confidentiality, anonymity, and integrity. However, neither transparency nor integrity is guaranteed in traditional voting processes, as they involve intermediaries who can manipulate the results.

Furthermore, despite facilitating voting procedures, online or electronic voting fails to ensure confidentiality, anonymity, or integrity because:

- Observers can monitor the procedure as it unfolds.
- The voting leaves a trace that can link each voter to their ballot.
- The voting system is centralized on a server controlled by an intermediary, allowing data to be modified.
- Web-based voting applications face server overload issues during the voting phase.

So, how can we avoid the aforementioned problems?

2.Proposed Solution

Our solution involves proposing an online voting system based on blockchain technology. Blockchain is an efficient, secure, and transparent means of managing the voting process.

We expect this system to:

- ✓ Ensure transparency: Voters themselves can verify all stages of an election, such as vote counting, to ensure that no votes have been deleted, manipulated, or modified.
- ✓ Ensure uniqueness: Each voter can cast one and only one vote.
- ✓ Ensure confidentiality: The voting procedure is not monitored during its execution, allowing voters to make their choices in secret.
- ✓ Ensure anonymity of voters: It is impossible to link a ballot to the voter who made the choice.
- ✓ Ensure security against potential fraud attempts at all levels.

Chapter 2: Analysis & Conception

I. Introduction

This chapter is dedicated to the fundamental steps of designing and modeling an online voting system based on blockchain technology. We have chosen the Unified Process (UP) as the design method and the Unified Modeling Language (UML) as the modeling language.

II. Unified Process (UP)

The Unified Process is an iterative and incremental software development process, built upon UML, centered around architecture and driven by use cases. It appears to be the ideal solution to address the perennial problem faced by developers. As illustrated in Figure 6 the UP approach can be summarized in the following steps:

- **Specification**: This step is used to define the various needs of the system:
    - Functional needs: From the user's perspective.
    - Non-functional needs: From a technical standpoint.
- **Analysis**: This step aims to understand the client's requirements and needs.
- **Design**: This step involves acquiring a coherent and comprehensive understanding of the system's architecture and structure.
- **Implementation**: This step focuses on translating the design into actual code.
- **Testing**: This step involves thoroughly testing the implemented code to ensure its quality and reliability.
- **Deployment**: This step encompasses the deployment of the developed system to the target environment.

- **Maintenance**: This step involves the ongoing maintenance and support of the system throughout its lifecycle.

The UP process provides a systematic approach to software development, ensuring that the needs and requirements of both users and stakeholders are effectively addressed throughout the development lifecycle.

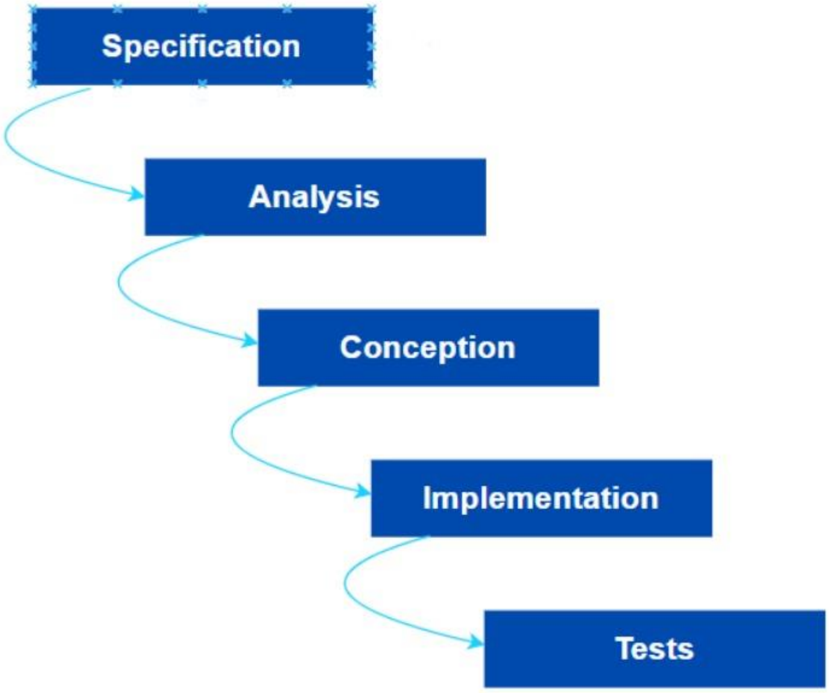


Figure 6 : Unified Process

III. Unified Modeling Language (UML)

UML (Unified Modeling Language) is a graphical and textual modeling language used to understand, describe requirements, specify and document systems, design solutions, and communicate perspectives. UML consists of thirteen diagrams. For the modeling of our system, we are using the following three fundamental diagrams:

- Use Case Diagram: This diagram expresses the behavior of the system in terms of actions and reactions from the perspective of each user. It defines the system's boundaries and its relationships with the environment.

- Class Diagram: This diagram represents the static structure of the system, showing the classes, their attributes, relationships, and methods. It provides an overview of the system's structure and helps in understanding the relationships between different components.
- Sequence Diagram: This diagram illustrates the interaction between various objects or components of the system over time. It shows the sequence of messages exchanged between the objects and the order in which they occur. It helps in understanding the dynamic behavior of the system [83].

By utilizing these UML diagrams, we can effectively model and communicate the different aspects and behaviors of our online voting system.

VI. Identification of Requirements

1. Functional Requirement:

The system should provide the following functionalities:

- ✓ Initiate the voting process.
- ✓ Authentication of voters and administrators.
- ✓ Addition of voters.
- ✓ Addition of candidates.
- ✓ Voting.
- ✓ Display of results.

2. Non-functional Requirements:

In addition to the fundamental needs, our system should meet the following criteria:

Traceability: The data should be securely recorded, timestamped, and sealed in a decentralized and tamper-proof ledger. This ensures that the data is certified and non-repudiable.

Security: The blockchain ensures that information is stored in an unmodifiable manner, and all this information can be chronologically accessed from a secure and easily accessible registry.

Performance: The software should be highly performant, meaning it should meet the requirements of users optimally through its functionalities.

Usability: The system should provide users with a simple and user-friendly interface.

Scalability: Users should be able to increase their processing, storage, transmission, and networking capabilities according to their needs.

By addressing these functional and non-functional requirements, our online voting system can provide a reliable, secure, and efficient platform for conducting elections.

V. Context Modeling :

1. Actors Identification and roles

An actor is an entity that defines the role played by a user or by a system that interacts with the modeled system. Here is a list of all actors appearing in use case diagrams.

Table 1: Organizer Roles

Actor	Roles
Organizer	1-Authenticate. 2-Manage Election: ✓ Add Election’s informations. -Set Election Title. -Set Election start time & end time. -View Election informations. ✓ Add Candidates informations. ✓ Search Candidates ✓ Reset the election. 3-Manage Voters: ✓ Import list of voters. ✓ Manage requests -Accept request. -Reject request. 4-Vote.

Table 2: Voter & User Roles

Actor	Roles
Voter	1-Authenticate. 2-Search Candidates 3-View Election details. -View Remaining time. -View Results. 4-vote.

User	1-Authenticate. 2-Register 3-View Election

2.1 Global Use Case Diagram

In this diagram we try to give a general idea concerning the users of this application by grouping them in a single figure in order to clarify their tasks and permissions.

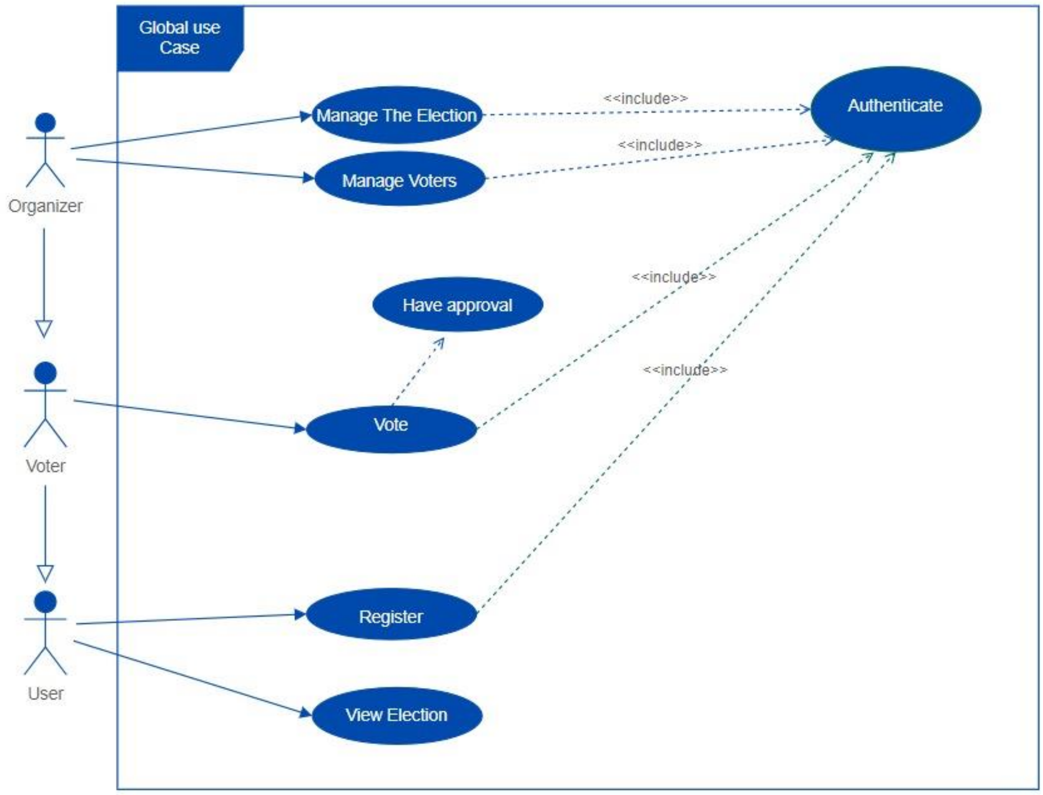


Diagram 1: Global Use Case Diagram

2.2 Organizer Use Case Diagram

Here, We observe that the main role of the Organizer is to manage the Election & Voters.

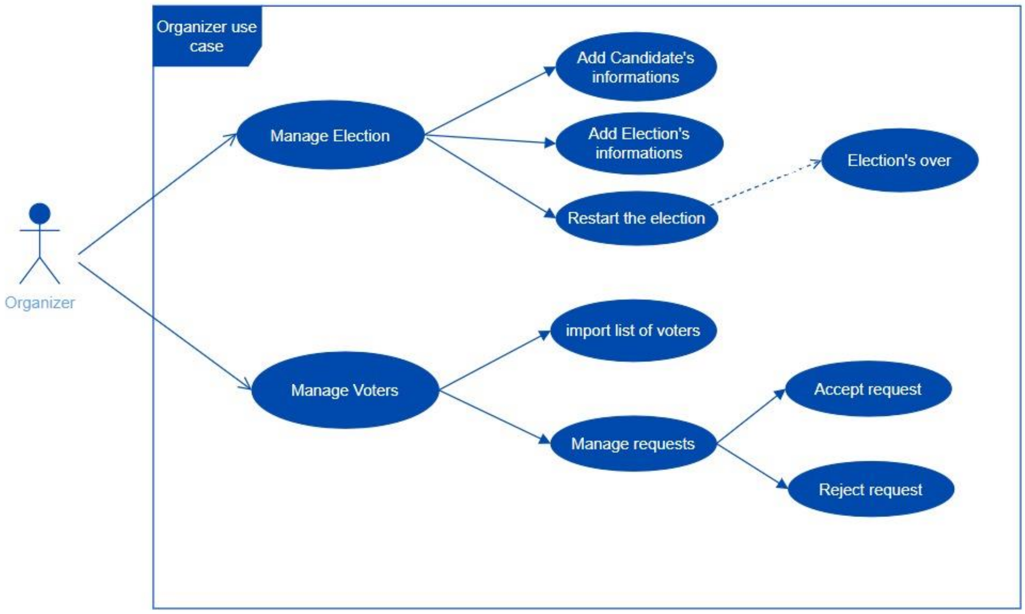


Diagram 2: Organizer Use Case Diagram

3.Sequence Diagrams

3.1 Login/Authentication Diagram

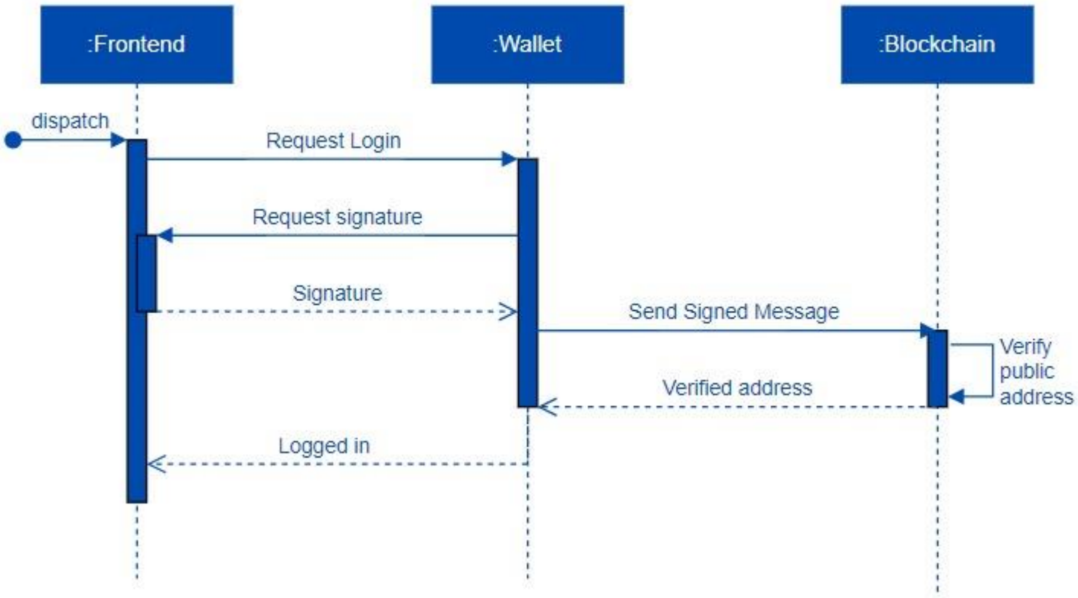


Diagram 3: Authentication Sequence Diagram

3.2 Voting Sequence Diagram

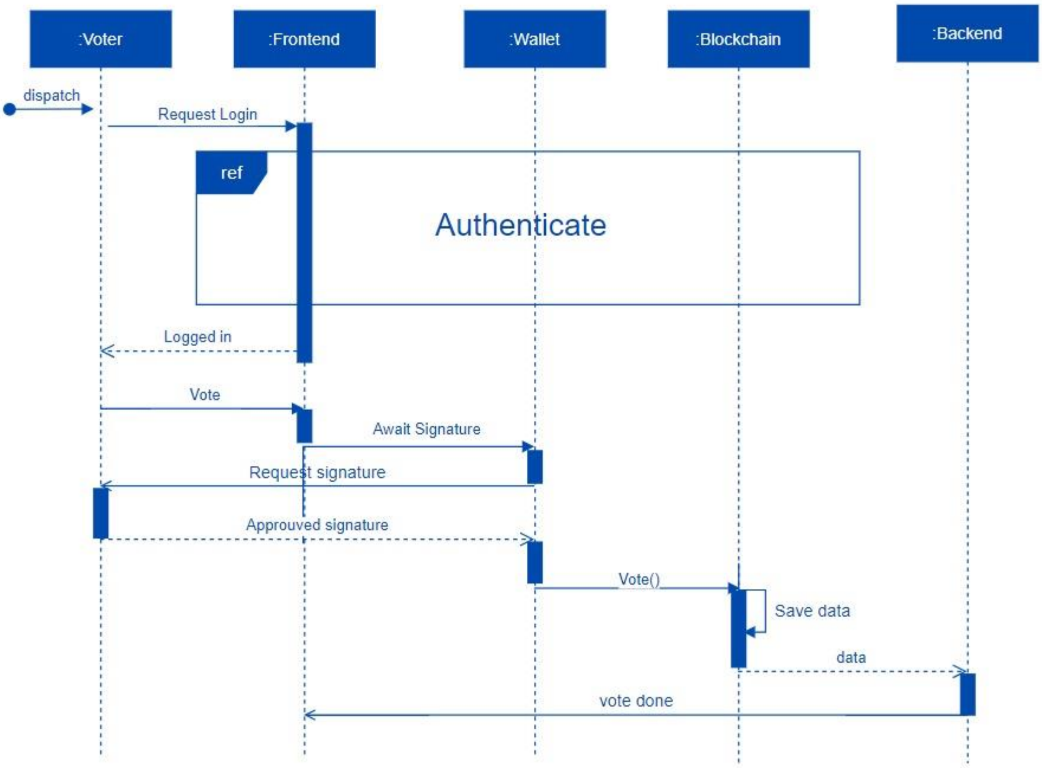


Diagram 4: Voting Sequence Diagram

These diagrams explain the processes to follow to submit a vote, how each entity interacts with the other one and how each event is arranged in time.

4. Smart Contract Conception Diagram

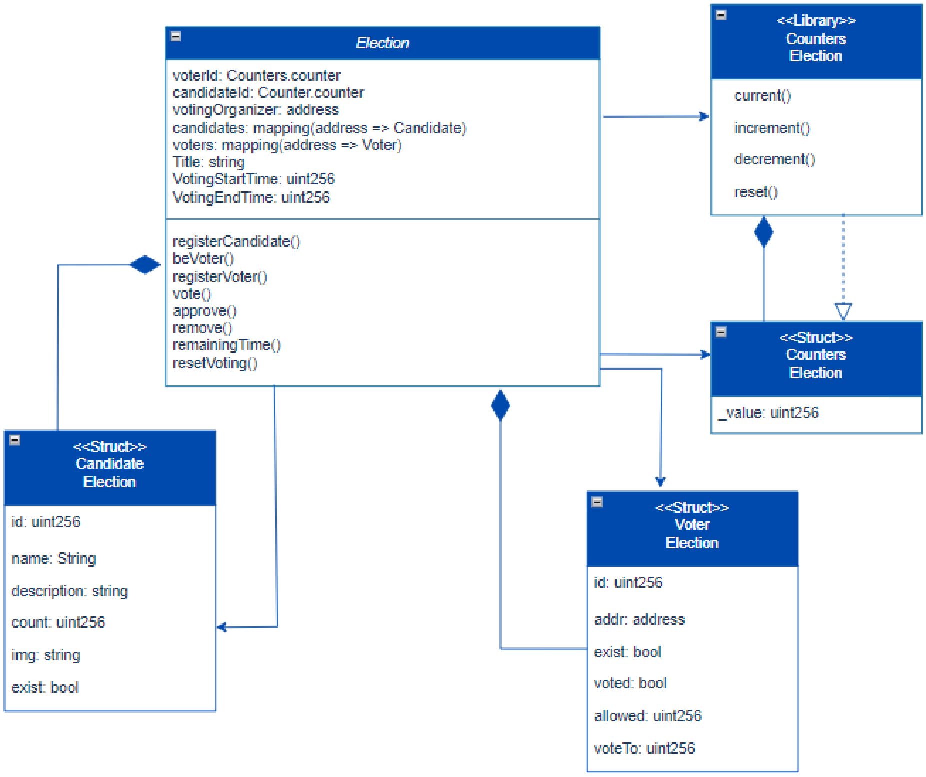


Diagram 5: Smart Contract Conception. (Generated By Remix IDE) Edited

The Election smart contract is designed to facilitate fair and transparent voting processes on the blockchain. The contract allows the organizer to create and manage candidates, register voters, set the start and end times for voting, and approve or remove voters. Once the voting period begins, registered voters can cast their votes for their preferred candidates. The contract ensures that voters can only vote once and that only approved voters have the right to vote. At the end of the voting period, the contract provides functions to retrieve comprehensive data on all voters and candidates, including the number of votes received by each candidate. The contract also allows the organizer to reset the voting process, clearing all candidate and voter data for future elections. By leveraging the transparency and immutability of the blockchain, the Election smart contract aims to enhance the integrity and efficiency of electoral processes.

Chapter 3: Project Realization

I. Introduction

In this chapter, we will first define all the programming languages, frameworks and tool we used to create our decentralized application.

II. Development tools and Technologies

1.Solidity

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behavior of accounts within the Ethereum state.

Solidity is a language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript.

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

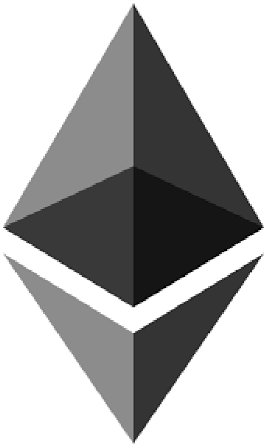
2.Remix IDE

Remix IDE is a web-based IDE (integrated development environment) that is used to develop, deploy, debug, and test Ethereum and EVM-compatible smart contracts. It is a powerful tool that can be used by developers at all levels of experience, and it does not require any setup. Remix IDE has a flexible, intuitive user interface, and it includes a rich set of plugins that can be used to extend its functionality.

3.Hardhat

Hardhat is a development environment to compile, deploy, test, and debug your Ethereum software. It helps developers manage and automate the recurring tasks that are inherent to the process of building smart contracts and DApps, as well as easily introducing more functionality around this workflow. This means compiling, running and testing smart contracts at the very core.

4.Ethereum (Goerli testnet)



Ethereum is a decentralized computing platform also known as Blockchain that uses ETH (also called Ether) to pay transaction fees (or “gas”). Developers can use Ethereum to run decentralized applications (DApps) and issue new crypto assets, known as Ethereum tokens.

5.metamask

Metamask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.

6.ipfs

The InterPlanetary File System is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices.

7.HTML/CSS/JavaScript

HTML, CSS, and JavaScript are the three fundamental technologies used to build and design websites and web applications. Here’s a brief overview of each technology:

HTML (The Hypertext Markup Language): is the standard markup language for documents designed to be displayed in a web browser.

It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript

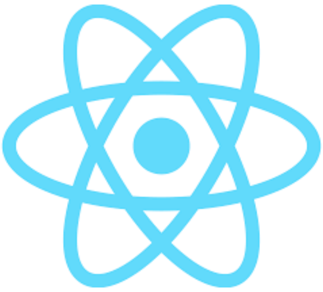
CSS (Cascading Style Sheets): is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

JavaScript: is an object-oriented computer programming language commonly used to add dynamic content to web pages, create interactive elements, respond to user events, access and manipulate data from the DOM (Document Object Model), which is the underlying structure of a web page.

8.TailwindCSS

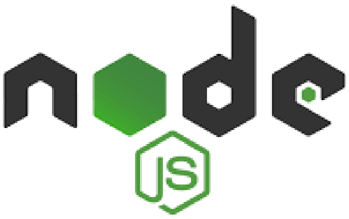
Tailwind CSS is a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.

9.React



React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

10.Node.js



Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

11.Thirdweb

Thirdweb is a complete web3 development framework that offers everything you need to connect your apps or games to decentralized networks.

With powerful tools and a large collection of ready-to-deploy contracts, thirdweb simplifies web3 development and allows you to easily build and deploy decentralized applications.

Also offers dashboards for everything, allowing you to easily manage and deploy your web3 solutions.

12.OpenZeppelin

OpenZeppelin is an open-source framework and a library of smart contracts for building secure and audited decentralized applications (dApps) on various blockchain platforms, primarily Ethereum. The framework provides developers with a set of standardized, tested, and community-audited smart contracts that can be used as building blocks for creating blockchain-based applications.

I. Work – Flow overview

This diagram includes the technologies used in every part of the development and more description of how each technology interacts with the other one.

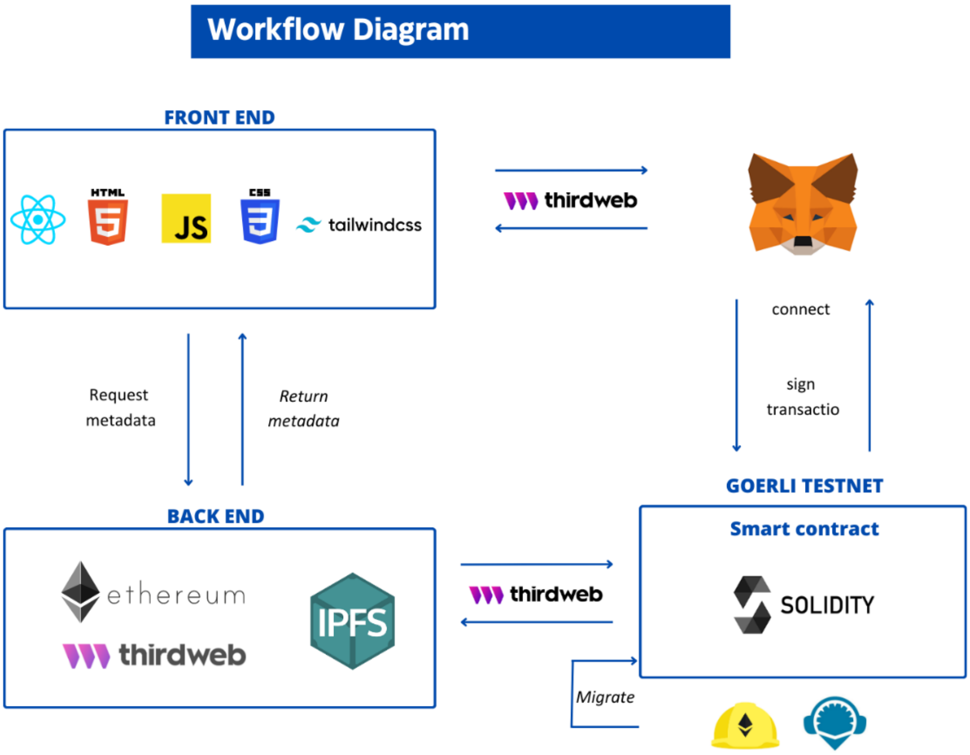
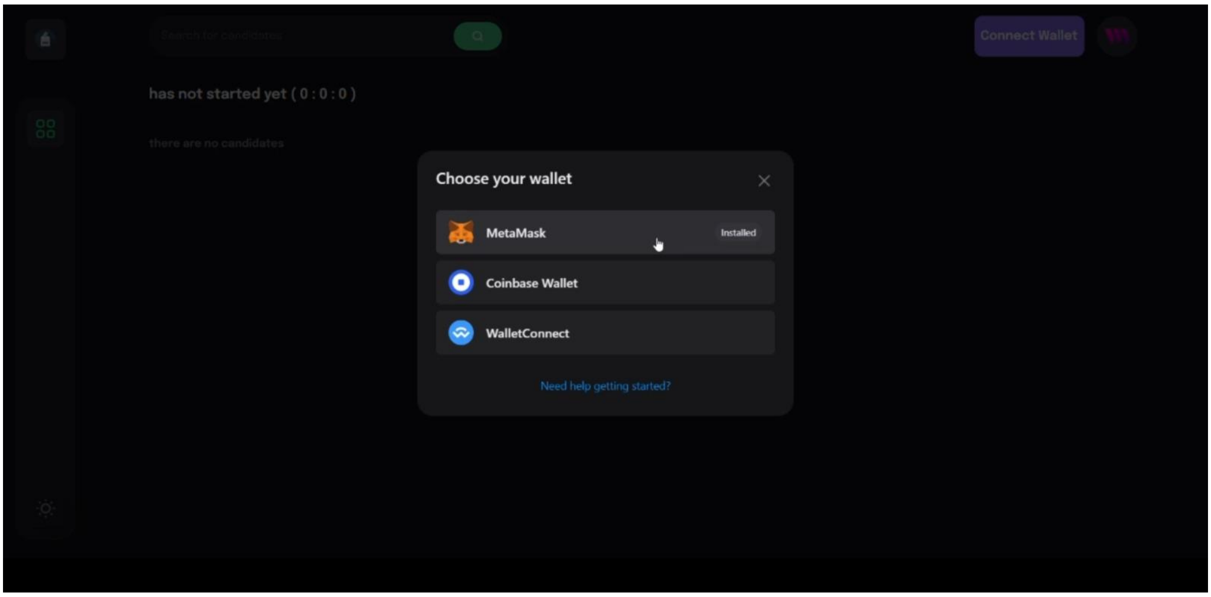


Figure 7 : Workflow diagram

IV. Project Presentation

1.Login Page

The first page is an invitation to connect the user's wallet if they are not already connected.

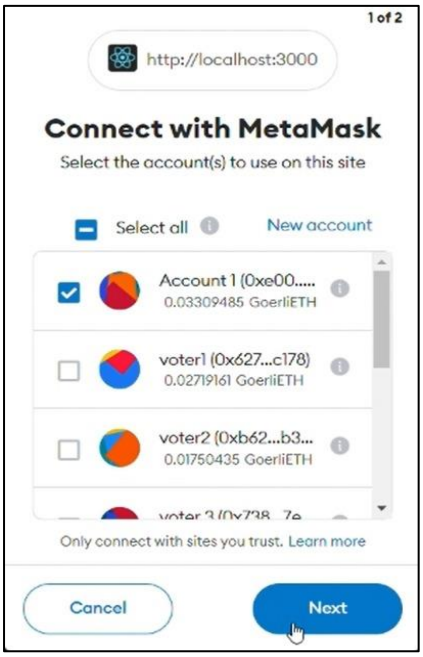


has not started yet (0 : 0 : 0)

there are no candidates

Choose your wallet

The connection process.



Connect with MetaMask

Select the account(s) to use on this site

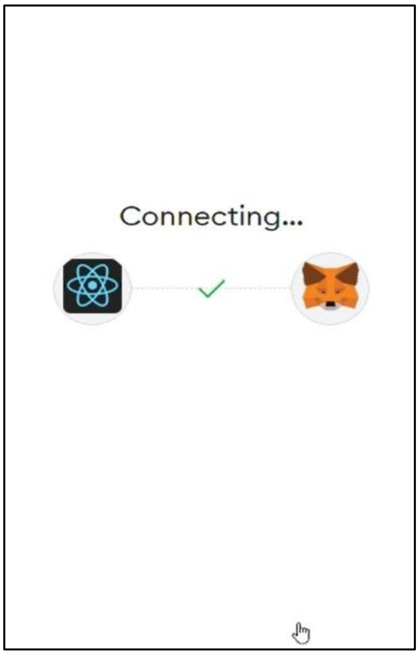
Account 1 (0xe00..... 0.03309485 GoerliETH

voter1 (0x627...c178) 0.02719161 GoerliETH

voter2 (0xb62...b3... 0.01750435 GoerliETH

voter 3 (0x738...7e

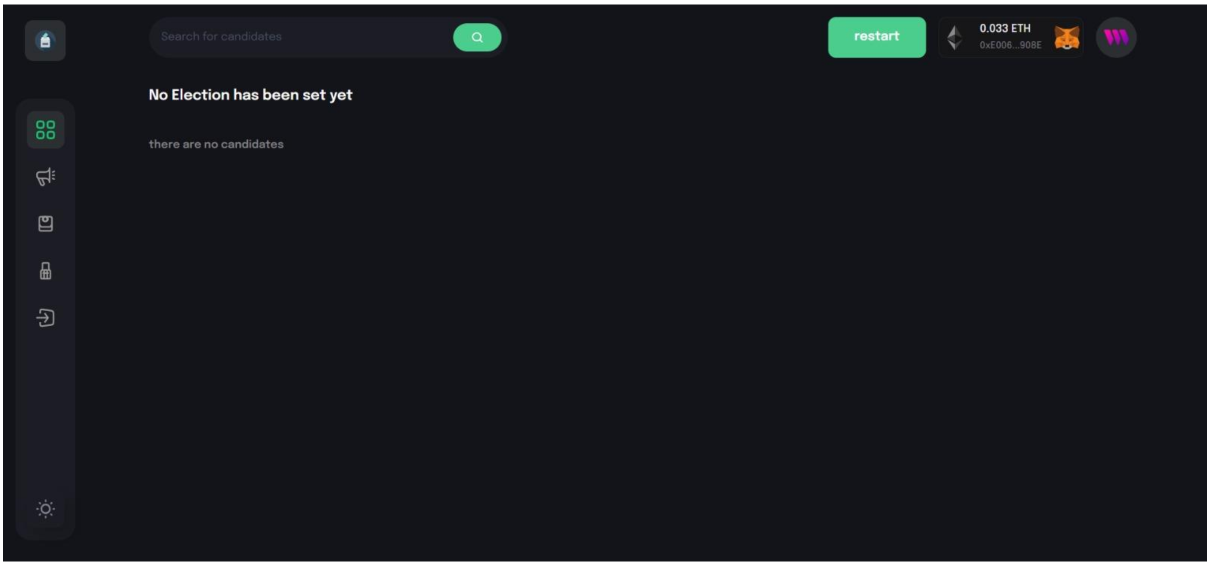
Only connect with sites you trust. Learn more



Connecting...

2.Home Page.

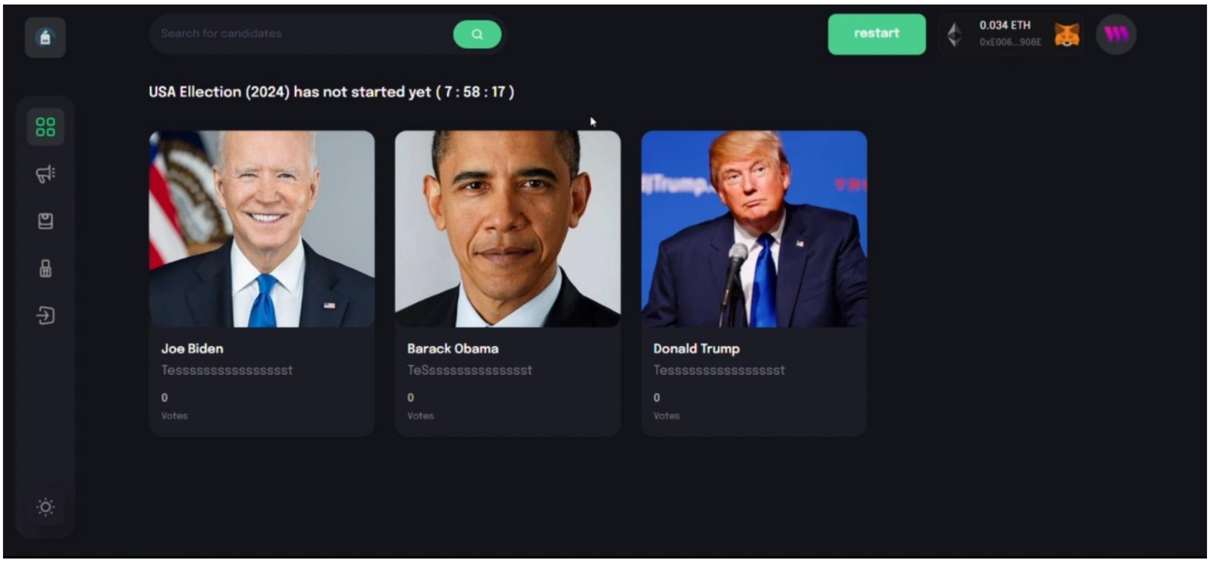
Home page when there is no Election going on which can be seen by all type of users



No Election has been set yet

there are no candidates

And with an actual election going on. providing all the candidates and their up-to-date votes, along with a countdown to the start and end of the election.



USA Ellection (2024) has not started yet (7 : 58 : 17)

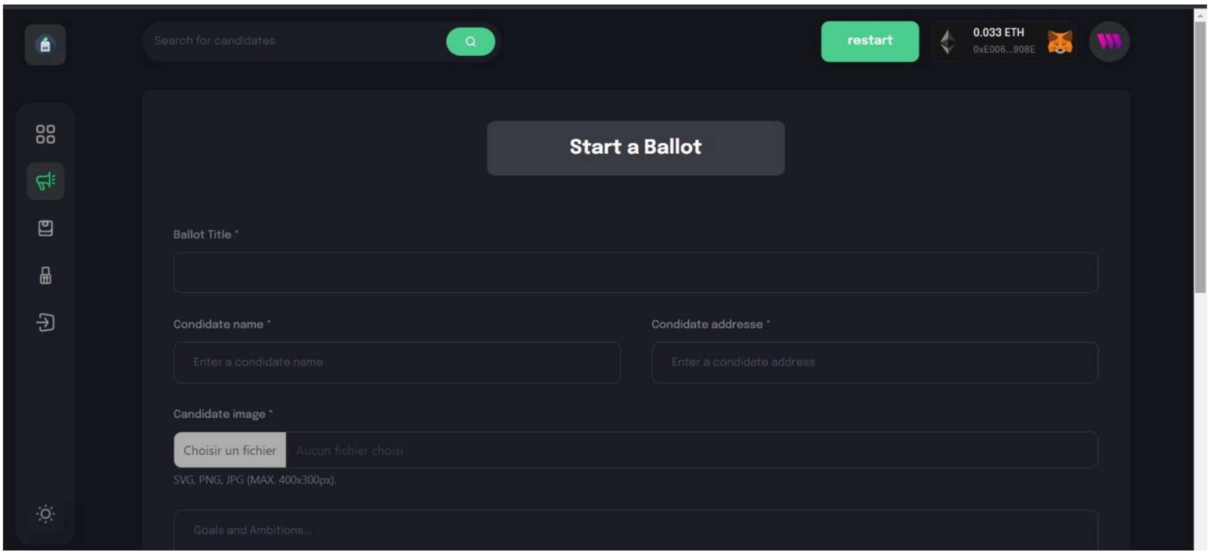
Joe Biden
Tesssssssssssssssst
0
Votes

Barack Obama
TeSssssssssssssssst
0
Votes

Donald Trump
Tesssssssssssssssst
0
Votes

3.Create Election Page.

The "Create Election" page can only be accessed and viewed by the organizer; this is where all the election information is provided.



Start a Ballot

Ballot Title *

Condidate name *

Enter a condidate name

Condidate addresse *

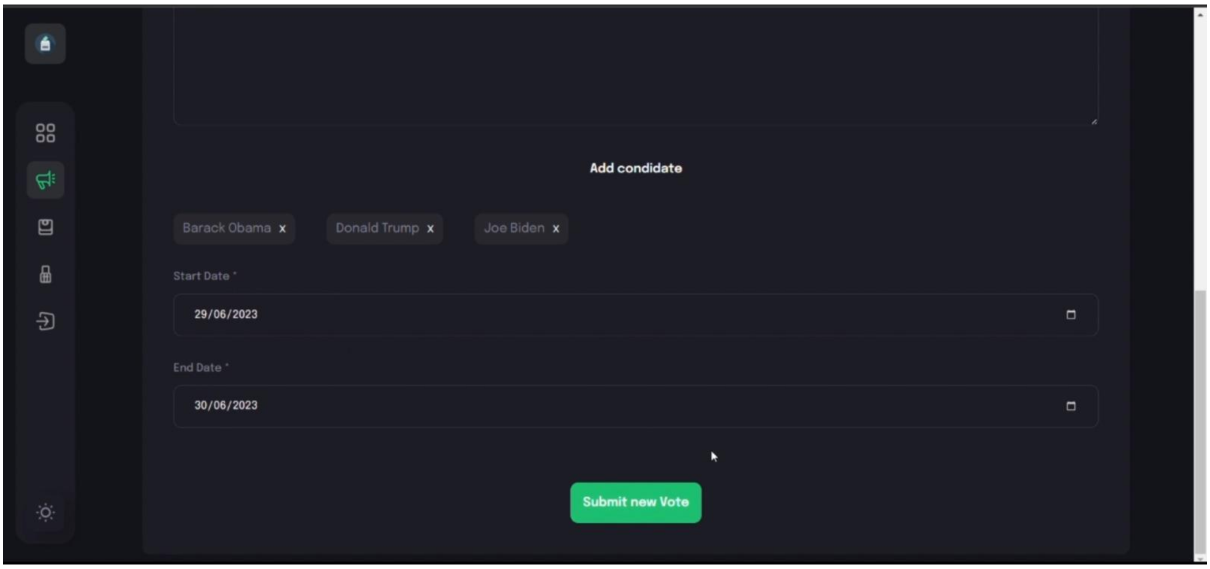
Enter a condidate address

Candidate image *

Choisir un fichier Aucun fichier choisi

SVG, PNG, JPG (MAX. 400x300px).

Goals and Ambitions...



Barack Obama x Donald Trump x Joe Biden x

Start Date *

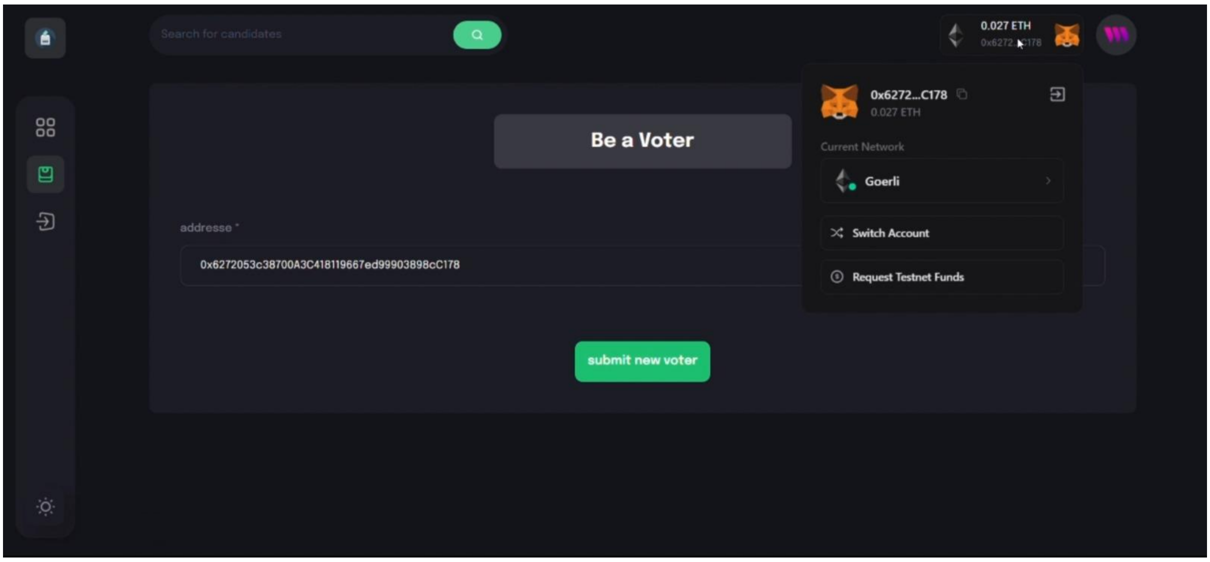
29/06/2023

End Date *

30/06/2023

4.Voter registration page.

“Voter registration” page is where the connected user has to apply in order to vote.

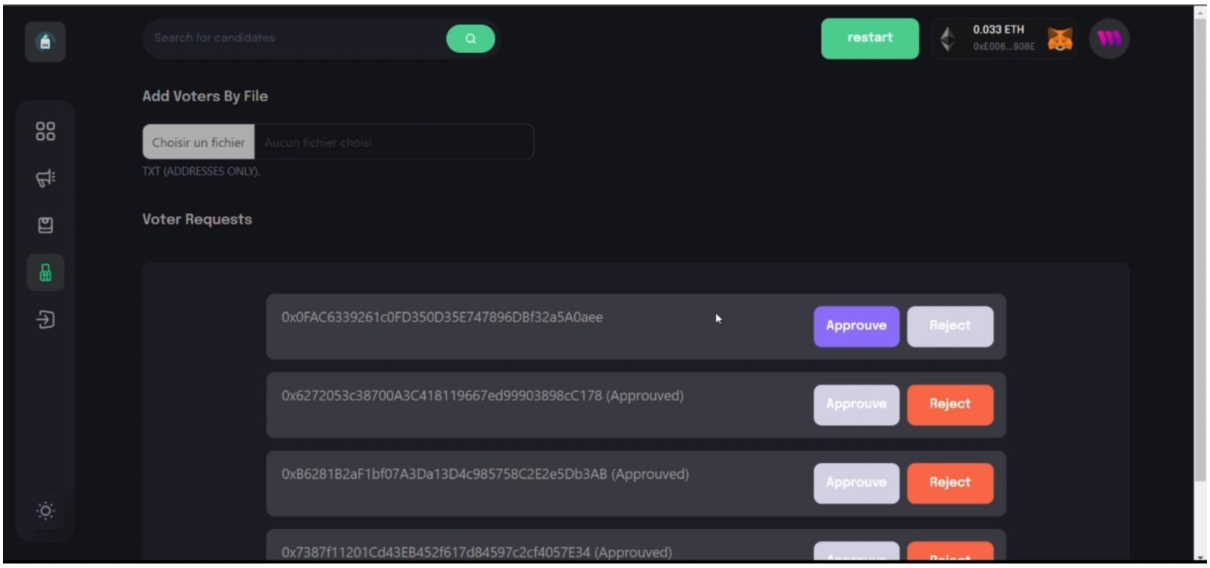


addresse *

0x6272053c38700A3C418119667ed99903898cC178

5.Voter approval page.

“Voter approval” page, used by the organizer in order to add the already allowed voters by a Text file of their addresses AND the request demands of the ordinary voters.



Add Voters By File

TXT (ADDRESSES ONLY).

Voter Requests

0x0FAC6339261c0FD350D35E747896DBf32a5A0aee Approuve Reject

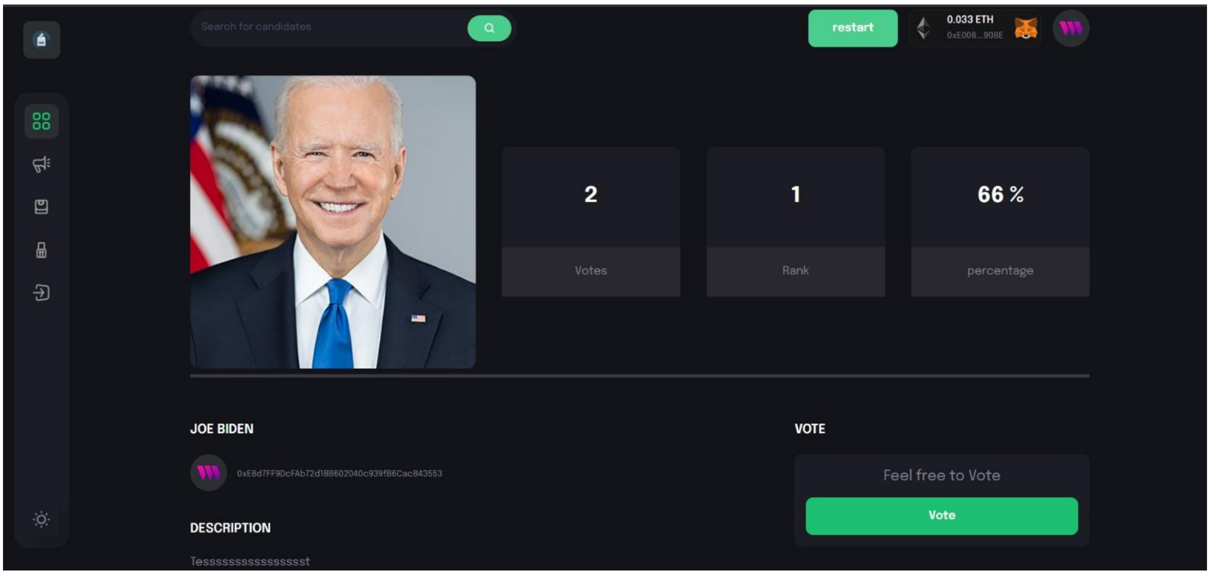
0x6272053c38700A3C418119667ed99903898cC178 (Approuved) Approuve Reject

0xB6281B2aF1bf07A3Da13D4c985758C2E2e5Db3AB (Approuved) Approuve Reject

0x7387f11201Cd43EB452f617d84597c2cf4057E34 (Approuved)

6.Candidate details.

Candidate details, filled with information about the candidate, also its where the voting option exists.



2
Votes

1
Rank

66 %
percentage

JOE BIDEN

0xE8d7FF9DcFAb72d1B8602040c939fB6Cac843553

DESCRIPTION

Tesssssssssssssssst

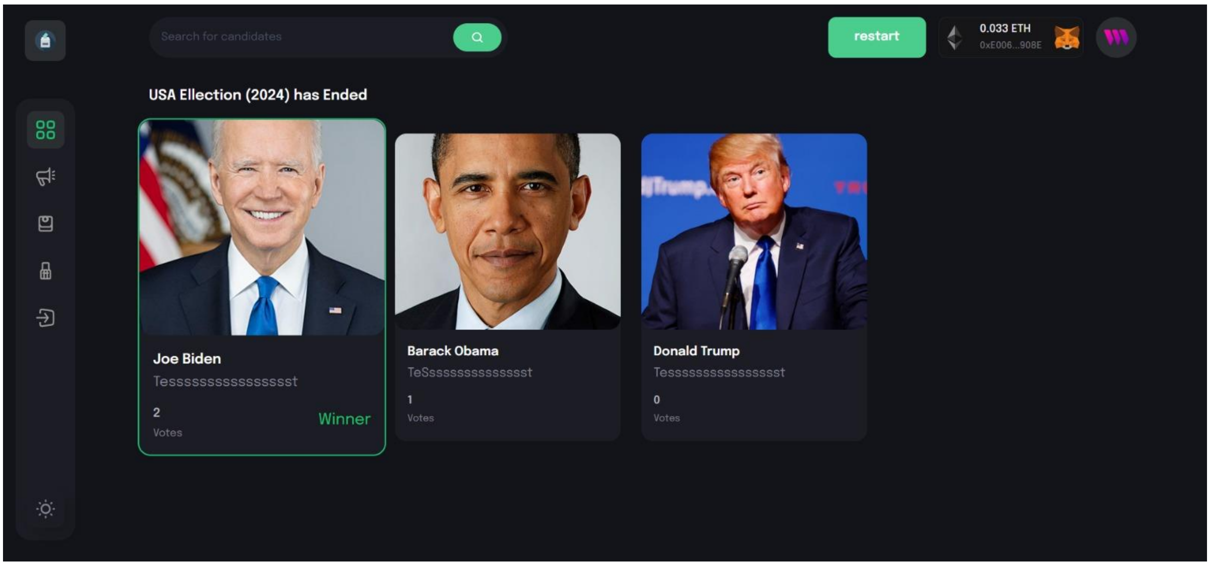
VOTE

Feel free to Vote

Vote

7.Winner Announcement.

Winner announcement when the election is over.



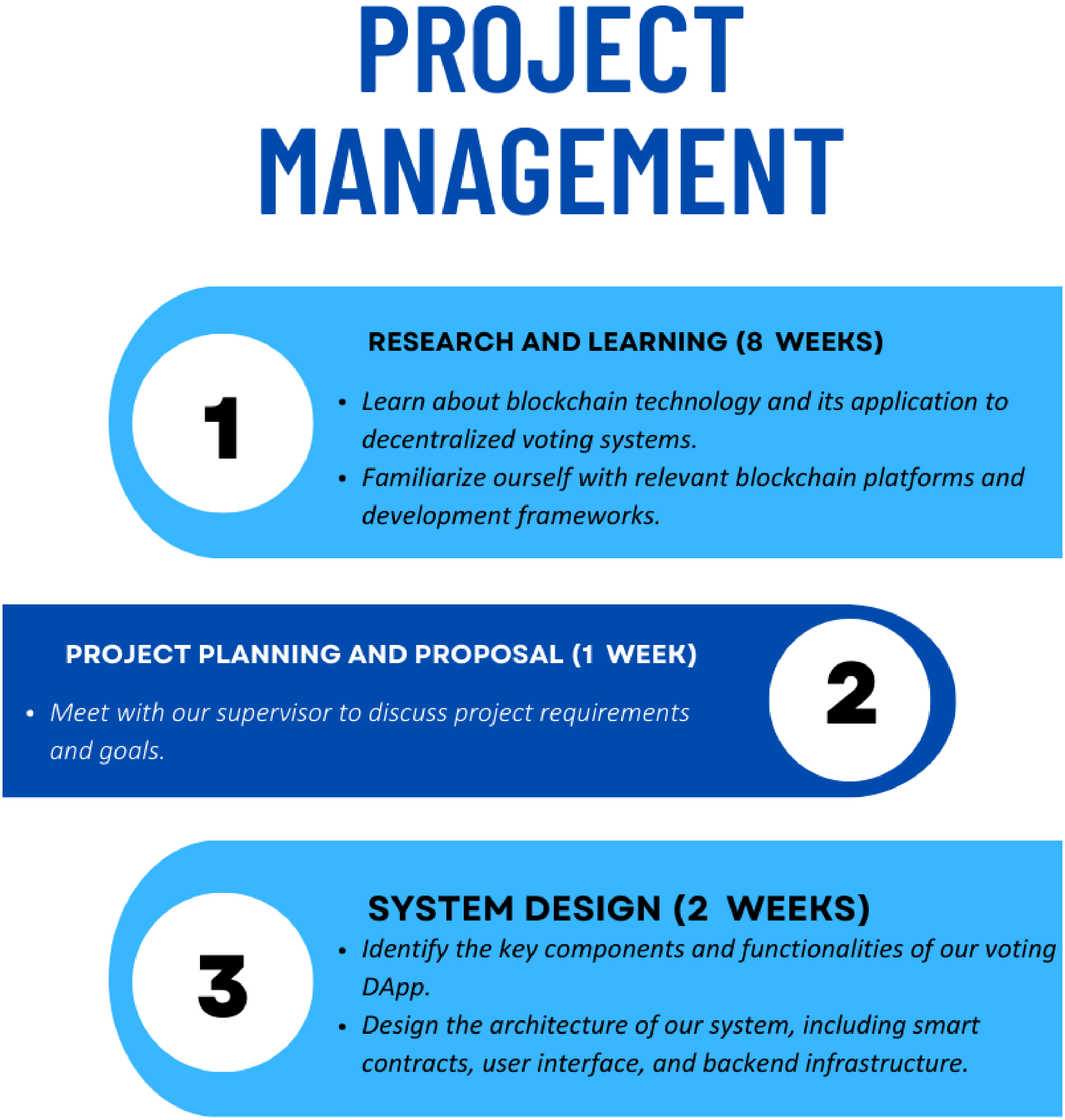
USA Ellection (2024) has Ended

Joe Biden
Tesssssssssssssssst
2
Votes
Winner

Barack Obama
TeSsssssssssssssst
1
Votes

Donald Trump
Tesssssssssssssssst
0
Votes

V. Project Management



PROJECT MANAGEMENT

1

RESEARCH AND LEARNING (8 WEEKS)

- *Learn about blockchain technology and its application to decentralized voting systems.*
- *Familiarize ourself with relevant blockchain platforms and development frameworks.*

2

PROJECT PLANNING AND PROPOSAL (1 WEEK)

- *Meet with our supervisor to discuss project requirements and goals.*

3

SYSTEM DESIGN (2 WEEKS)

- *Identify the key components and functionalities of our voting DApp.*
- *Design the architecture of our system, including smart contracts, user interface, and backend infrastructure.*

DEVELOPMENT (10 WEEKS)

4

- *Set up the development environment and necessary tools.*
- *Implement the smart contracts and business logic of the voting DApp.*
- *Develop the frontend user interface, ensuring usability and responsiveness.*
- *Integrate the frontend with the backend and smart contracts.*

TESTING AND QUALITY ASSURANCE (3 WEEKS)

5

- *Conduct unit testing to ensure the functionality of individual components.*
- *Conduct user acceptance testing with a group of test accounts.*
- *Identify and fix any bugs or issues.*

DOCUMENTATION AND FINALIZATION (1 WEEK)

6

- Document the project's technical details, including system architecture.
- Finalize the project report, including a summary of the research, design decisions, implementation details.

PRESENTATION AND SUBMISSION (1 WEEK)

7

- Prepare a presentation to showcase our voting DApp and its features.
- Present our project to our supervisor.

I. Summary & Conclusion

1. Summary of Achievement

By accomplishing this project, we gained a lot of knowledge about the solidity programming language and basic concept of decentralized network (Block-chain). Not only we acquire a knowledge of technical aspect but also the importance of planning and scheduling of the project. we have created a frontend of election results where we successfully implemented and tested the contract of migration and election. It can Test the various condition at Remix IDE where it ensure that contract code is bug free. If the contract contains any bugs, it might disable the contract and deploy the new copy. It minimizes the cost of ether during the transaction. Setup the thidweb and inject the metamask to the local server. Each account (voters) by linking the private key at the Ethereum we can vote only one time. We can view the unique address of the account(voter) after migrating into the blockchain. Successfully incremented vote count of the candidate which lead to change in the state of the block chain.

2.Future Work

While the current version of the election dApp system has achieved significant milestones, there are several areas that could be explored for future improvements and enhancements.

Here are some potential avenues for future work:

Enhanced Security Measures: Continuously strengthening the security measures of the dApp system should be a priority.

Exploring advanced encryption techniques, multi-factor authentication, and secure wallet integration can further enhance the security and prevent any potential vulnerabilities.

Integration with External Systems: Consider integrating the dApp system with external systems such as government identity verification databases or existing voter registration systems.

This integration would streamline the voter approval process and ensure the accuracy of voter identities.

Batch smart contract operations, in order to reduce financial and computational costs.

Scalability and Performance: As the system gains atraction and more elections are conducted, it is essential to ensure that the dApp can handle increased traffic and a growing number of participants.

Optimizing the system for scalability and performance will be crucial for its long-term success.

Collaboration with Election Authorities: Collaborate with relevant election authorities and regulatory bodies to ensure compliance with legal requirements and incorporate their expertise in the development and improvement of the dApp system.

By addressing these areas of future work, the voting dApp system can continue to evolve and adapt to the changing needs and challenges of the electoral process. It has the potential to revolutionize the way elections are conducted, ensuring transparency, security, and trust in democratic processes.

Webography: