BlockBallot

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by
Gaurang Mittal(20204075)
Gagan Gupta(20204074)
Gulla Guna Harsha(20204078)
Adnan Hassan(20204015)

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD PRAYAGRAJ
May, 2023

UNDERTAKING

I declare that the work presented in this report titled "*BlockBallot*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

May, 2023
Allahabad

(Gaurang Mittal(20204075)
Gagan Gupta(20204074)
Gulla Guna Harsha(20204078)
Adnan Hassan(20204015))

CERTIFICATE

Certified that the work contained in the report titled "*Block-Ballot*", by
Gaurang Mittal(20204075)
Gagan Gupta(20204074)
Gulla Guna Harsha(20204078)
Adnan Hassan(20204015),
has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Professor Rajesh Tripathi)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

May, 2023

Preface

Titled "**BlockBallot**", is a team project to develop a decentralized voting system.The motivation behind this project was to explore the potential of blockchain technology in addressing some of the challenges faced by traditional voting systems, such as voter fraud, manipulation, and lack of transparency. By creating a decentralized voting application, we aimed to build a system that would enable secure, transparent, and tamper-proof voting, without the need for intermediaries or trusted third parties.

In this report, we describe the design, development, and evaluation of our decentralized voting application. We first discuss the background and related work in the field of blockchain-based voting systems. We then present the architecture and implementation details of our voting application, including the use of smart contracts, cryptographic protocols, and consensus mechanisms. We also describe the user interface and functionality of the application, as well as the deployment and testing process.

Overall, this project report showcases our team's effort and learning in designing and building a decentralized application using blockchain technology. We hope that this report will contribute to the growing body of knowledge on blockchain-based voting systems and inspire further innovation in this field.

Acknowledgements

I would like to express my sincere gratitude to my project advisor, **Professor Rajesh Tripathi**, for their guidance and support throughout the development of this project. Their inputs were instrumental in shaping the direction of my research.I would also like to thank my classmates and colleagues for their help and encouragement. I am grateful for their willingness to share their knowledge and expertise with me.

Finally, I would like to express my appreciation to the MNNIT Allahabad, which provided me with the amazing and competitive environment to develop this project.

Contents

Chapter 1

Introduction

Voting is a fundamental process in democratic societies, allowing citizens to participate in the decision-making process and elect their representatives. However, traditional voting systems are often centralized, meaning that they are controlled by a single authority and can be vulnerable to tampering and fraud. Decentralized voting applications, on the other hand, offer a potential solution to these issues by leveraging decentralized technologies such as blockchain and distributed ledger technology.

This report describes the development of a decentralized voting application designed to increase transparency, security, and accessibility in the voting process. The application utilizes blockchain technology to provide a tamper-proof and auditable voting system, allowing voters to cast their votes securely and anonymously. The system is designed to be user-friendly and accessible to all eligible voters, regardless of their location or technical knowledge.

In this report, we will discuss the design and implementation of the decentralized voting application, including the choice of technology, the security mechanisms implemented, and the user interface. We will also examine the potential benefits and challenges of decentralized voting applications and discuss the legal and regulatory considerations that need to be taken into account.

1.1 Problems in current systems

The current voting system has several problems, including centralized control, lack of transparency, limited accessibility, security vulnerabilities, and long wait times. These issues can potentially compromise the integrity and accuracy of the voting process, leading to allegations of fraud, reduced voter turnout, and a lack of trust in the democratic process.

1. **Lack of Transparency**: The current voting system often lacks transparency, making it difficult to verify the accuracy and integrity of the election results. The process of vote counting and result declaration is often opaque, and voters are not able to see how their vote is counted or who is responsible for the counting process.

2. **Security Concerns**: The current voting system is vulnerable to various security threats such as hacking, tampering, and manipulation. Malicious actors can manipulate voting machines, compromise voter registration databases, or engage in other forms of cyberattacks to influence the election results.

3. **Centralization**: The current voting system is often centralized, with a few entities controlling the entire process. This centralization can create opportunities for corruption, fraud, and abuse of power, undermining the fairness and credibility of the election.

4 .**Accessibility**: The current voting system can be inaccessible to certain groups of voters, such as those with disabilities, the elderly, or those living in remote areas. This can result in voter disenfranchisement and undermine the principles of democratic representation.

5. **Cost and Time**: The current voting system can be costly and time-consuming, requiring significant resources to manage and maintain. This can lead to long wait times for voters, as well as delays in counting and reporting the results.

6. **Paper-Based Processes**: The current voting system often relies on paper-based processes, which can be inefficient, error-prone, and susceptible to loss or damage. This can lead to inaccuracies in the vote counting process and undermine the integrity of the election.

1.2 Motivation

The need for a secure, transparent, and decentralized voting system has become increasingly urgent in recent years. The current voting system is plagued with various problems as stated in previous section. In addition, recent controversies surrounding traditional voting methods, such as allegations of foreign interference in elections and vote manipulation, have undermined public trust in the democratic process.

The BlockBallot project is motivated by the need to develop a decentralized voting application that leverages the benefits of blockchain technology to create a more secure, transparent, and trustworthy voting system. The project aims to address the limitations and challenges of the current voting system, and provide a viable alternative that enables direct participation by voters, while also ensuring the accuracy and integrity of the election results.

The development of the BlockBallot application represents an important contribution to the field of blockchain-based voting systems, and has the potential to transform the way we conduct elections and uphold the principles of democratic representation. The project is motivated by a vision of a more transparent, secure, and decentralized world, where individuals can exercise their democratic rights and participate in the governance of their communities and societies.

Chapter 2

Technical Backgrounds

In this chapter, we provide an introduction to the technical theory behind blockchain technology, with a specific focus on the Ethereum framework used to build our minimum viable decentralized voting system. We explain the fundamental concepts and terms related to blockchain technology, as well as the technical details necessary to understand the underlying technology.

2.1 Blockchain

A blockchain is a revolutionary technology that has transformed the way data is recorded, shared, and verified in a decentralized and secure manner. It is essentially a digital ledger that allows transactions to be recorded in a transparent, tamper-resistant, and immutable way across a network of computers. Blockchains are characterized by their decentralized and distributed nature, which eliminates the need for a central authority or intermediary to oversee transactions.

At its core, a blockchain relies on complex cryptographic algorithms to ensure data integrity and security. Transactions are recorded in blocks, which are then linked together to form a chain of information that cannot be altered or deleted once added to the blockchain. Each block contains a unique digital signature, known as a hash, that is generated based on the information it contains and the hash of the previous block. This linking of blocks ensures that any attempt to tamper with

one block would invalidate all subsequent blocks, making it virtually impossible to manipulate or corrupt the data.

The potential applications of blockchain technology are vast and varied, ranging from financial transactions and supply chain management to voting systems and smart contracts. In addition to its security and transparency benefits, blockchains can also enhance efficiency, reduce costs, and improve accountability in a range of industries and use cases. Despite its transformative potential, however, the technology is still in its early stages of development and requires further innovation and adoption to realize its full potential.

Information on the blockchain is transparent and anyone can view the content of a blockchain. Transactions on the blockchain are not completely anonymous. However, information about the users is limited to their digital signature.

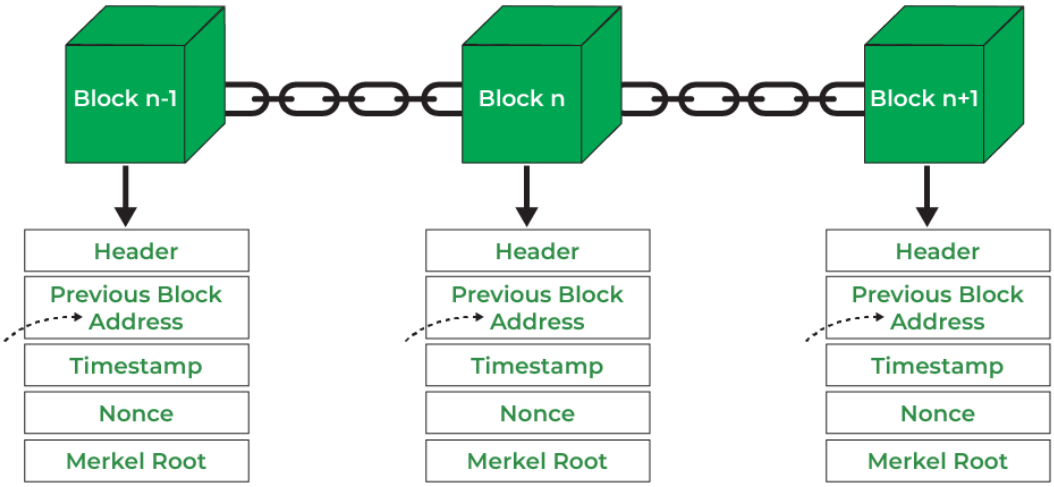


Figure 1

2.1.1 Transactions

A transaction in a blockchain refers to a set of instructions that modify the blockchain's state. Public blockchain networks require transaction fees to be paid in order for specialized nodes, also known as miners, to process and confirm the transactions.

In Bitcoin, transactions facilitate the transfer of value, and a user's account

balance is determined by the sum of all individual Unspent Transaction Outputs (UTXOs) associated with their private key. Other blockchain networks, such as Ethereum, offer more complex functionality by providing a transaction-based state machine. This enables developers to create a range of applications that rely on blockchain technology, leveraging the transaction-based state machine concept to achieve various functionalities and use cases.

2.1.2 Digital Wallets

In a blockchain network, each user is assigned a unique asymmetric key pair consisting of a public key and a private key. The public key is openly available to anyone and serves as the user's digital wallet address, with no connection to the user's identity. Conversely, the private key is kept secret and allows the owner of the wallet to sign valid transactions.

For a transaction to be considered valid, it must be signed with the corresponding private key, generating a digital signature that makes it virtually impossible to tamper with or duplicate. The transaction's digital signature serves as a unique identifier, ensuring that only the wallet's owner can authorize a specific transaction.

By employing asymmetric key cryptography, blockchain technology provides a secure and decentralized means of conducting transactions, eliminating the need for intermediaries or centralized authorities. The public nature of blockchain networks and the anonymity afforded by digital wallet addresses further enhance the privacy and security of users' transactions.

2.1.3 Miners

Miners in blockchain networks are specialized nodes that perform the complex computations required to validate and confirm transactions on the network. These computations typically involve solving cryptographic puzzles and require significant computational power.

Miners are essential to the functioning of blockchain networks, as they are responsible for adding new blocks to the chain and ensuring the integrity and immutability

of the ledger. In most cases, miners are incentivized to participate in the network through rewards, such as cryptocurrency or transaction fees.

2.1.4 Consensus Mechanism

In a blockchain network, a consensus mechanism refers to the process by which the network's nodes reach an agreement on the current state of the ledger. This agreement is critical to ensuring the integrity and immutability of the blockchain.

Consensus mechanisms work by requiring network participants to validate and confirm transactions through a complex and often computationally intensive process. In most cases, this involves solving complex cryptographic puzzles, with the first participant to solve the puzzle being rewarded for their efforts.

One common consensus mechanism used in many blockchain networks is called Proof of Work (PoW). PoW requires participants to solve a cryptographic puzzle, with the difficulty of the puzzle increasing as the network grows. This mechanism ensures that the network remains secure and resistant to attack.

Another consensus mechanism that has gained popularity in recent years is called Proof of Stake (PoS). Unlike PoW, which requires significant computational power, PoS requires participants to hold a certain amount of the blockchain's native cryptocurrency. The more cryptocurrency a participant holds, the more likely they are to be selected to validate the next block of transactions.

Consensus mechanisms are critical to the functioning of blockchain networks, as they ensure that all network participants agree on the current state of the ledger, preventing double-spending and other fraudulent activities.
Some of the most common consensus mechanisms include: Proof of Work, Proof of State, Byzantine Fault Tolerance and Proof of Authority.

2.2 Ethereum

Ethereum is a decentralized, open-source blockchain platform that enables developers to build and deploy decentralized applications, or dapps. It was created as an evolution of the Bitcoin blockchain, with a focus on enabling more complex and

sophisticated smart contracts and decentralized applications.

Unlike Bitcoin, which is primarily designed for peer-to-peer transactions, Ethereum is a programmable blockchain that allows developers to build custom applications on top of the blockchain itself. These applications can be designed to automate complex business processes, create decentralized finance platforms, build decentralized social networks, and much more.

One of the key features of Ethereum is its support for smart contracts, which are self-executing contracts that are programmed to automatically enforce the terms of an agreement. Smart contracts can be used to automate a wide range of business processes, from supply chain management to digital identity verification.Ethereum also has its own native cryptocurrency, Ether (ETH), which is used to pay for transaction fees and other costs associated with using the platform. Like other cryptocurrencies, the value of Ether is determined by supply and demand on cryptocurrency exchanges.

2.2.1 Smart Contracts

Smart contracts are self-executing digital contracts that are stored on a blockchain network and automatically enforce the terms of the agreement between parties. Smart contracts are typically used on blockchain networks such as Ethereum, which allow for the creation of decentralized applications (DApps) that can execute smart contracts. These contracts are written in a programming language called Solidity, which allows for the creation of complex conditions and rules that can be enforced automatically.

2.2.2 Solidity

Solidity is a high-level programming language used for writing smart contracts on the Ethereum blockchain. It is designed to enable developers to create complex applications that can run on the decentralized network by defining the rules for executing transactions. Solidity is a statically-typed language that supports inheritance, libraries, and user-defined types among other features, making it a flexible

tool for creating custom contracts. It is also designed to be secure, preventing common programming mistakes and vulnerabilities that could lead to the loss of funds or other issues. Overall, Solidity is a crucial tool for building decentralized applications that can be trusted to function as intended on the Ethereum network.

2.2.3 Gas

Gas is used to pay for the execution of smart contracts on the Ethereum network. Gas is a fee paid in Ether, the native cryptocurrency of the Ethereum platform. Gas prices can fluctuate depending on network activity and the complexity of the smart contract.

Chapter 3

Proposed Work

1. We developed a web application that is integrated with blockchain technology, specifically the Ethereum blockchain. The purpose of the system is to provide a decentralized voting platform that is secure, transparent, and tamper-proof.

2. The BlockBallot provides a registration section for voters in which they need to enter their basic details such as name, email address and password to be a user on the BlockBallot.

3. When the administration starts the voting process, the users need to login in the BlockBallot and register themselves as a voter by providing aadhar number and Ethereum account address to caste their votes. At this stage Aadhar authentication will be done via OTP on Aadhar linked mobile number and the user will only be registered as a voter if the user's age is 18 and above.

4. The BlockBallots to register themselves and securely cast their votes using their digital identities. The system stores the votes on the Ethereum blockchain using smart contracts, which are executed automatically and without the need for a centralized authority. This ensures that the voting process is transparent and cannot be tampered with.

5. The proposed system also includes a web-based dashboard for administrators to manage the voting process. The dashboard allows administrators to create

and manage the voting events, add candidates, and track the voting results in real-time.

The visual representation of our proposed model is described in figure 2.

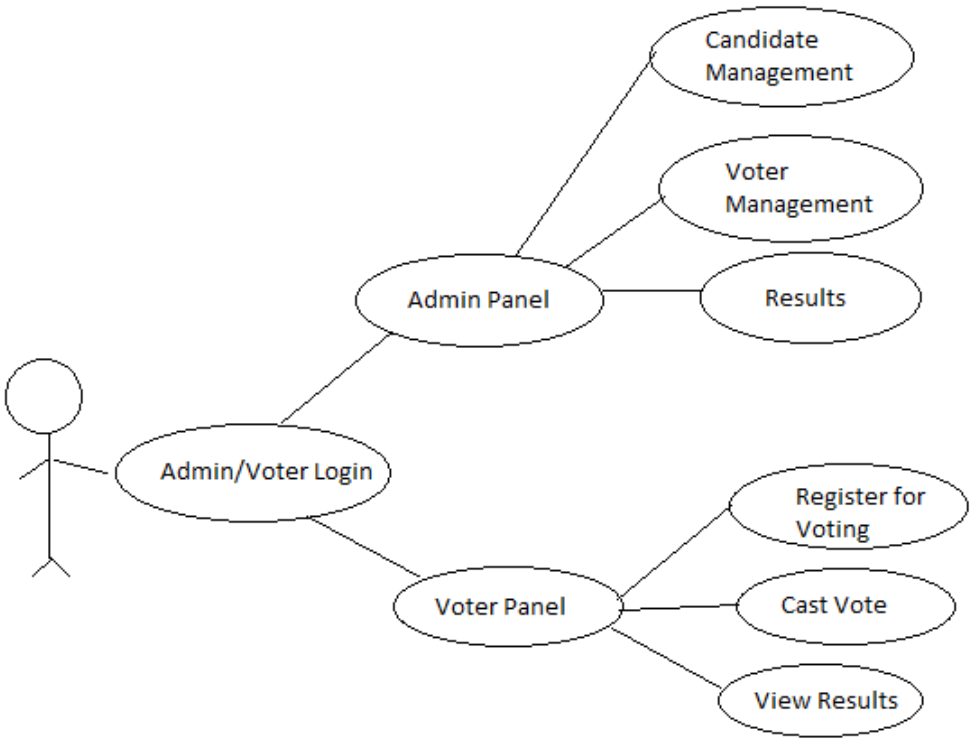


Figure 2: Use case diagram

Chapter 4

Tech Stack

1. **HTML**: Hypertext Markup Language (HTML) is a widely used markup language that is used to structure content on the web. It was used in BlockBallot to create the web application's structure.

2. **CSS**: Cascading Style Sheets (CSS) is a language used to describe how a web page should be presented. It was used in BlockBallot to define the visual elements of the web application, such as typography, colors, and layout.

3. **Bootstrap**: Bootstrap is a front-end framework used for building responsive web applications. It provides pre-built CSS and JavaScript components as well as a responsive grid layout system. It was used in BlockBallot to create a responsive layout and style.

4. **Node.js**: Node.js is a server-side JavaScript runtime environment that allows developers to create efficient and scalable web applications. It was used to build the back-end of the BlockBallot web application.

5. **Express.js**: Express.js is a Node.js web application framework that provides robust features for building web applications and APIs. It was used in BlockBallot to create the server-side application logic.

6. **EJS**: EJS (Embedded JavaScript) is a templating engine used to generate HTML markup with JavaScript. It was used to generate dynamic HTML

pages for the BlockBallot web application, allowing us to display dynamic data to the user.

7. **MySQL**: MySQL is an open-source relational database management system used for storing and managing data in a structured way. It was used to store user information and voting data securely.

8. **Ganache**: Ganache is a locally hosted blockchain designed for development and testing, allowing developers to deploy and test smart contracts on a local network. It was used to test and deploy the smart contracts utilized in the BlockBallot application.

9. **Truffle**: Truffle is a development framework used to build, test, and deploy smart contracts on the Ethereum blockchain. It was used to develop and test the smart contracts used in the BlockBallot application.

10. **Metamask**: Metamask is a browser extension that allows users to interact with the Ethereum blockchain from their web browser. It was used to allow users to sign transactions and vote on the BlockBallot application.

11. **Ethereum**: Ethereum is a decentralized blockchain platform that allows developers to build and deploy decentralized applications. It was used as the underlying blockchain technology for the BlockBallot application, providing a secure, transparent, and tamper-proof voting system.

In summary, our tech stacks included HTML, CSS, Bootstrap, ExpressJs, Node.js, EJS, MySQL, Ganache, Truffle, Metamask, and Ethereum. The combination of these technologies allowed us to create a secure, transparent, and tamper-proof decentralized voting application.

Chapter 5

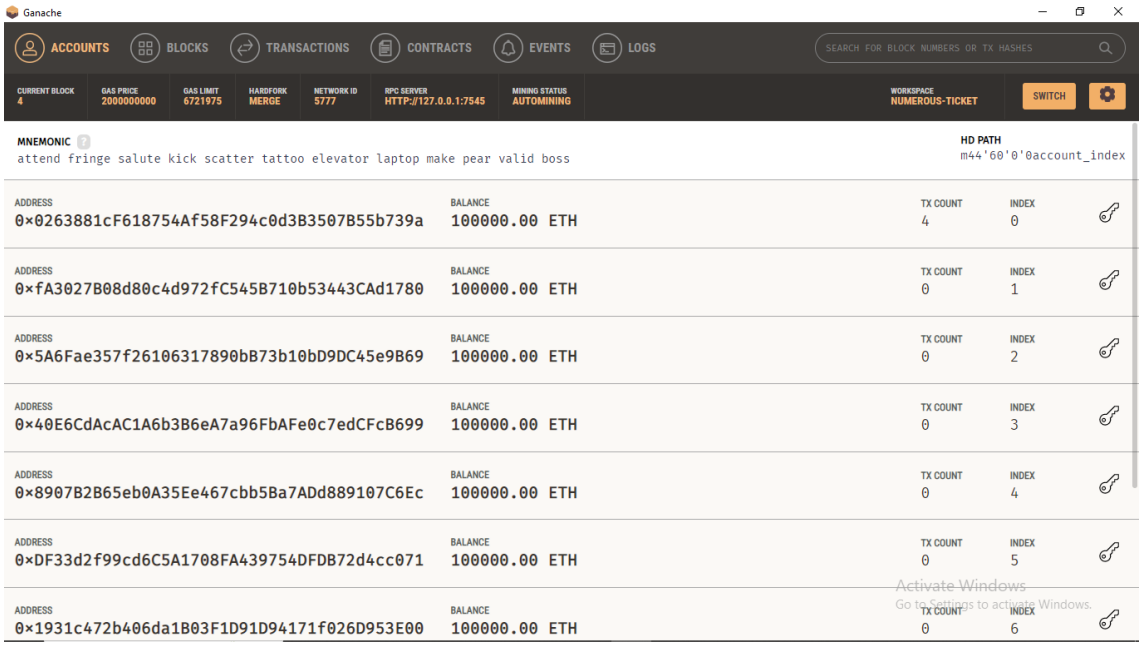
Project Setup and Experimental Results

5.1 Overview

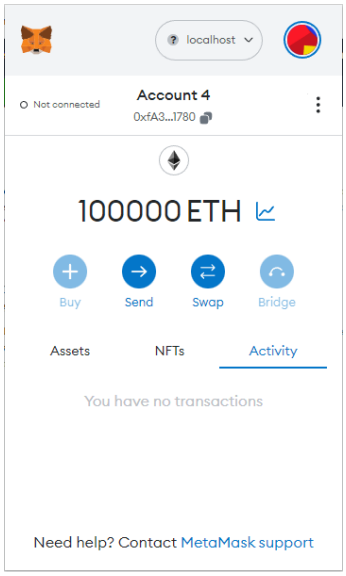
Till now, we have explored the blockchain technology and the proposed work through the web application BlockBallot. Now it's time to set it up on your local machine too. There are few steps to be followed to make it work.

5.2 Steps for project setup

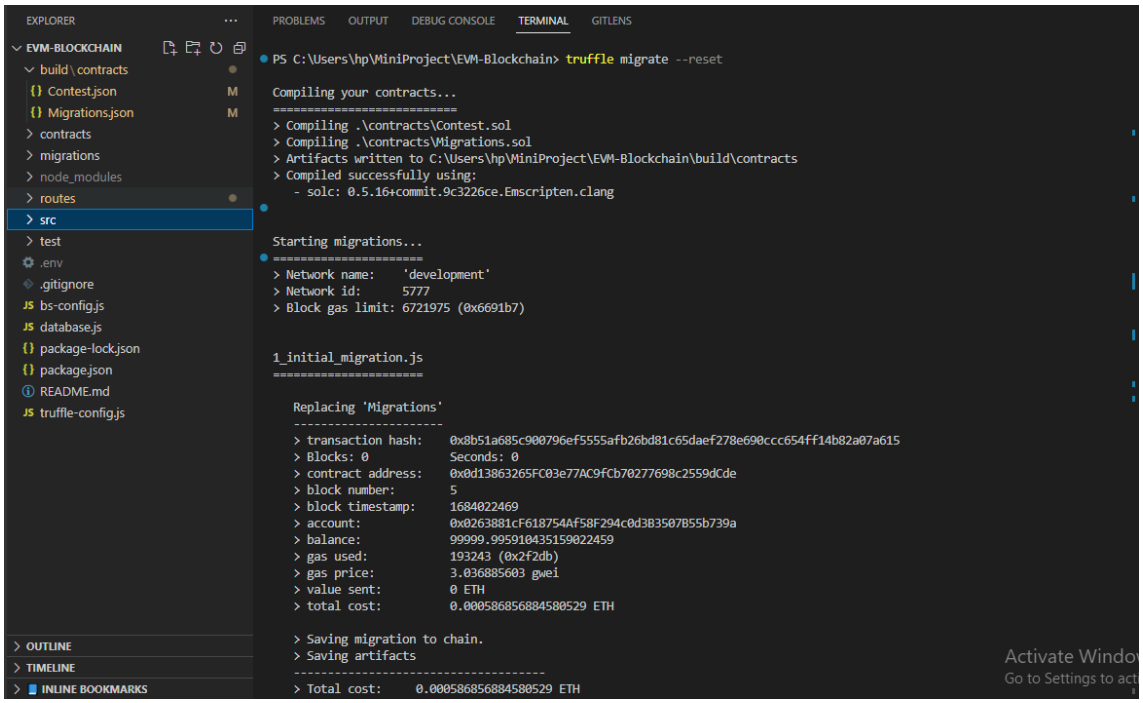
- Clone this given repository by typing
  git clone https://github.com/Gagan1729-droid/EVM-Blockchain.git
- Type the below into your terminal to install all modules and dependencies which BlockBallot is using:
  npm install
- Install the Ganache client which provides 10 ethereum accounts with 100 ethers on localhost. You can customize it. Add the project folder to link with the truffle-config.js file.



- Install the Metamask wallet, and import the account from the ganache client. Create a local network with following configurations:
  Network Name: localhost
  RPC Url: http://localhost:7545
  Chain Id: 1377
  Currency Symbol: ETH



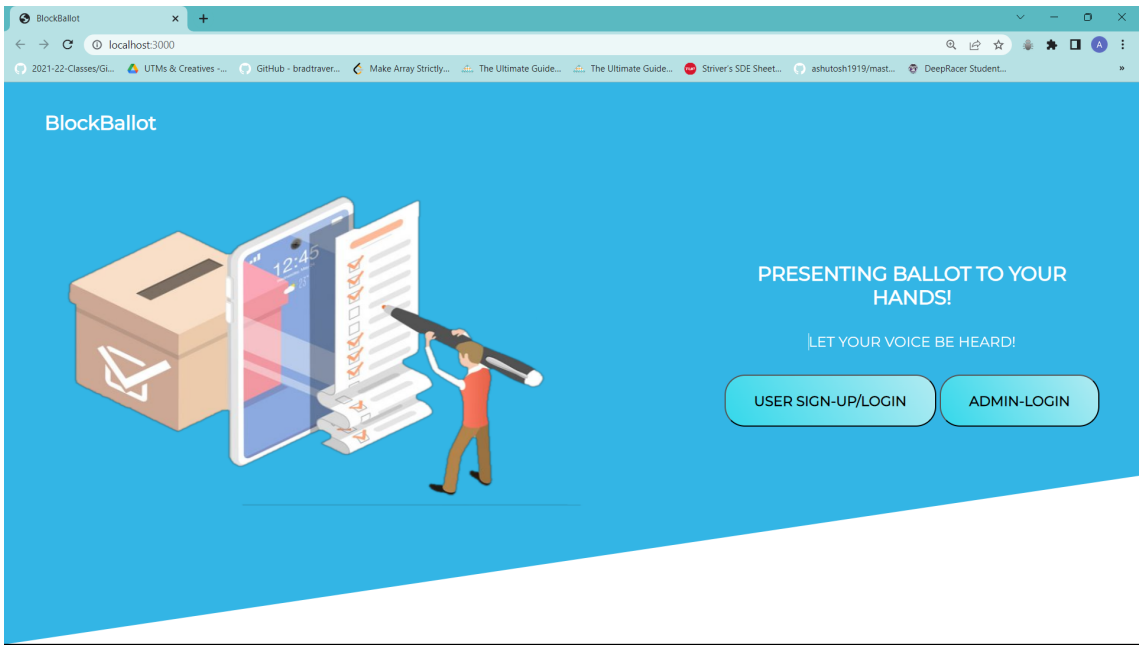
- Open the folder with project files and run command :
  truffle migrate —reset
  This command compiles our smart contract and deploys it to initiate the chain.
  Also the first account in Ganache is set as the administrator account.



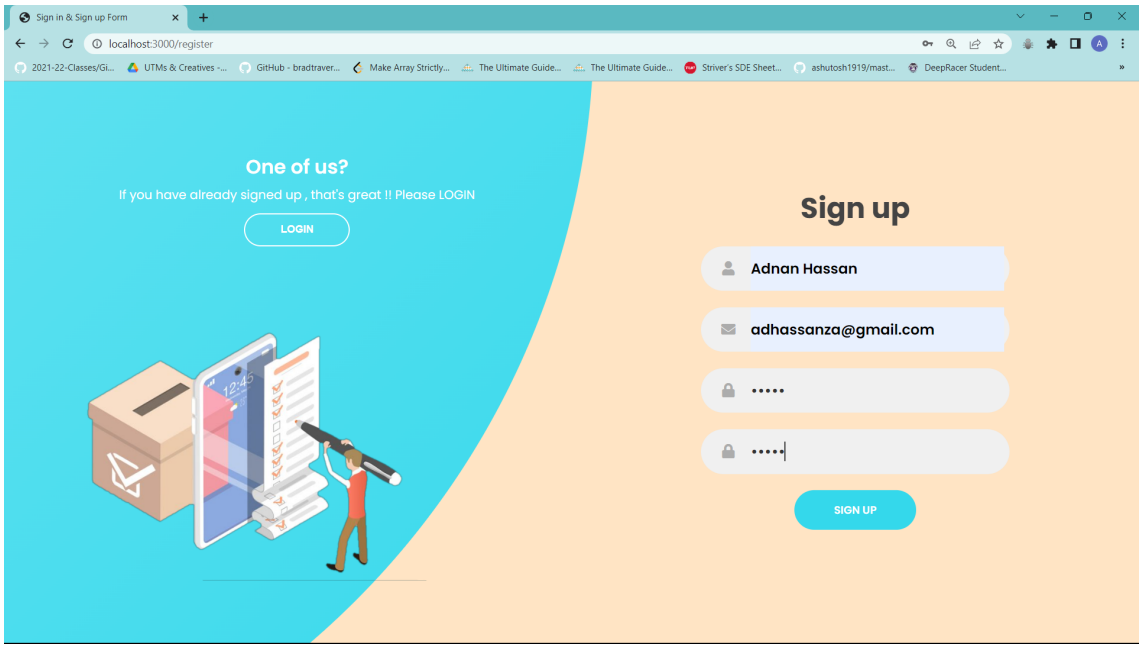
- Now run the command in the terminal to start our web application:
  npm start

5.3 Experimental Results

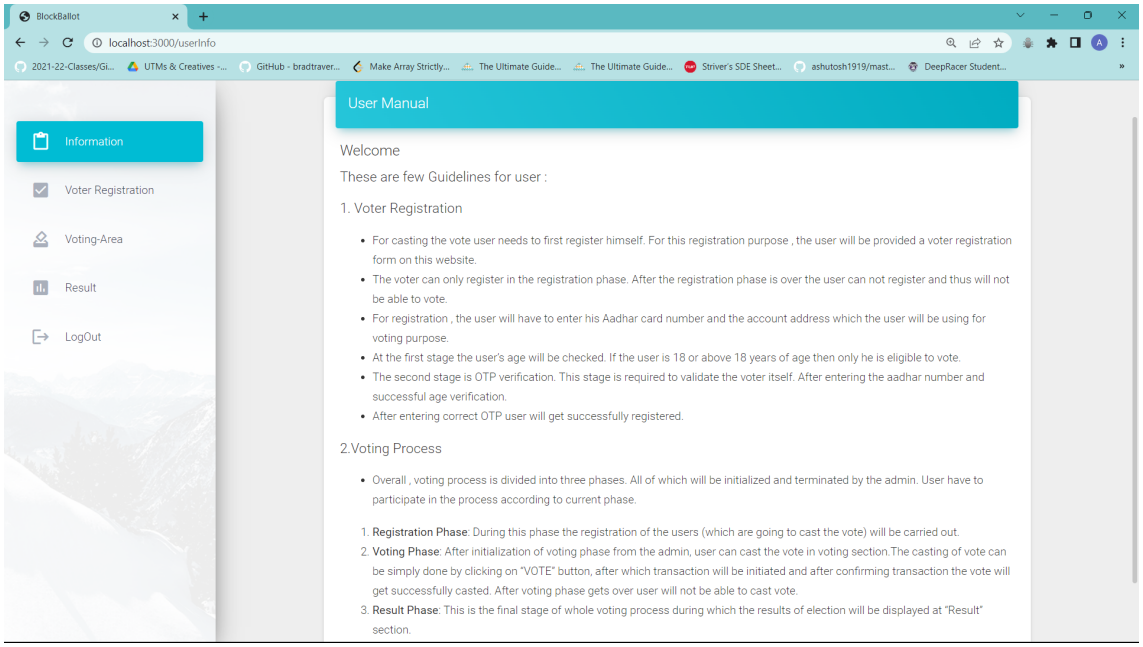
- Start the project and go to localhost:3000 in the browser. You will see the following screen.



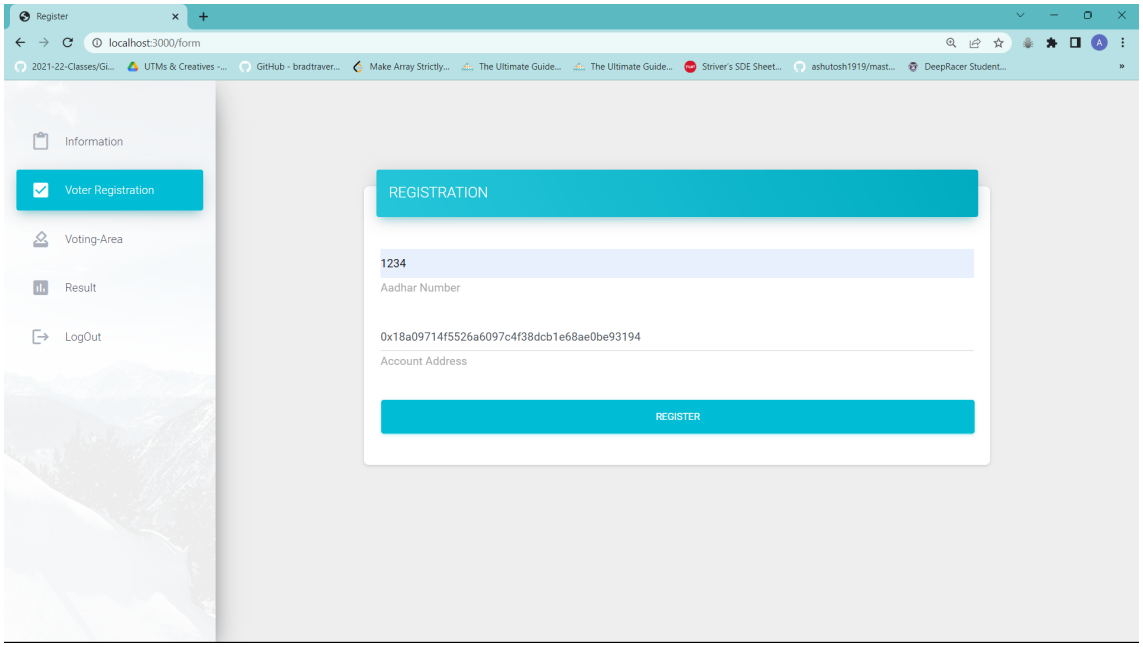
- Move to the sign-up part of the user. Enter the credentials in the form and register.



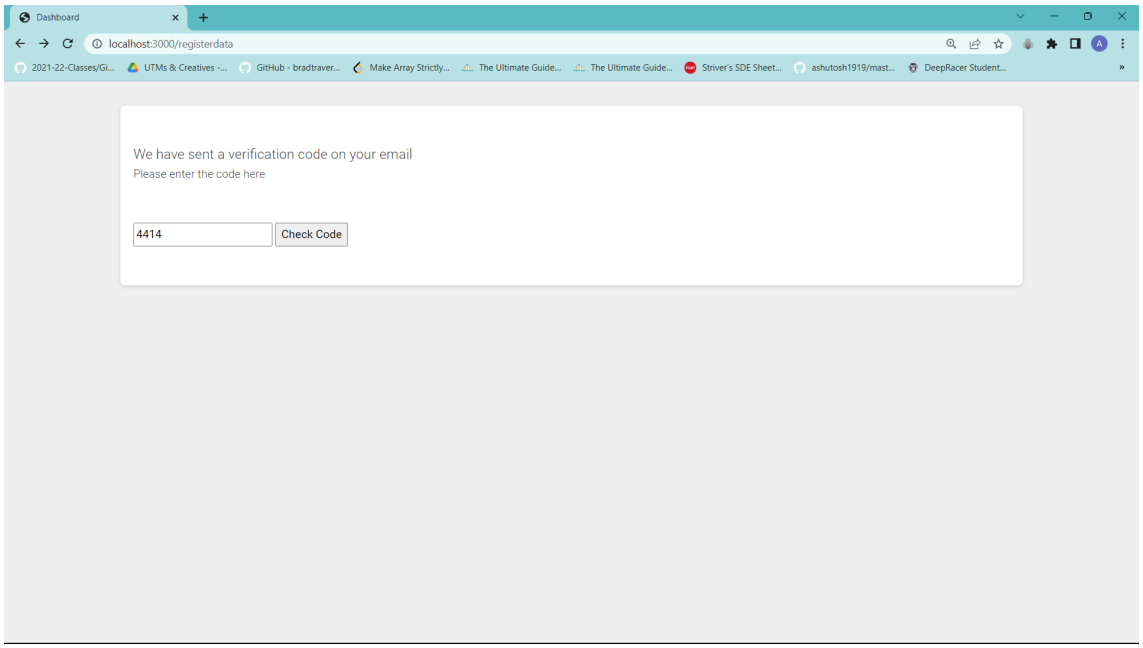
- The dashboard menu appears where we see all the instructions



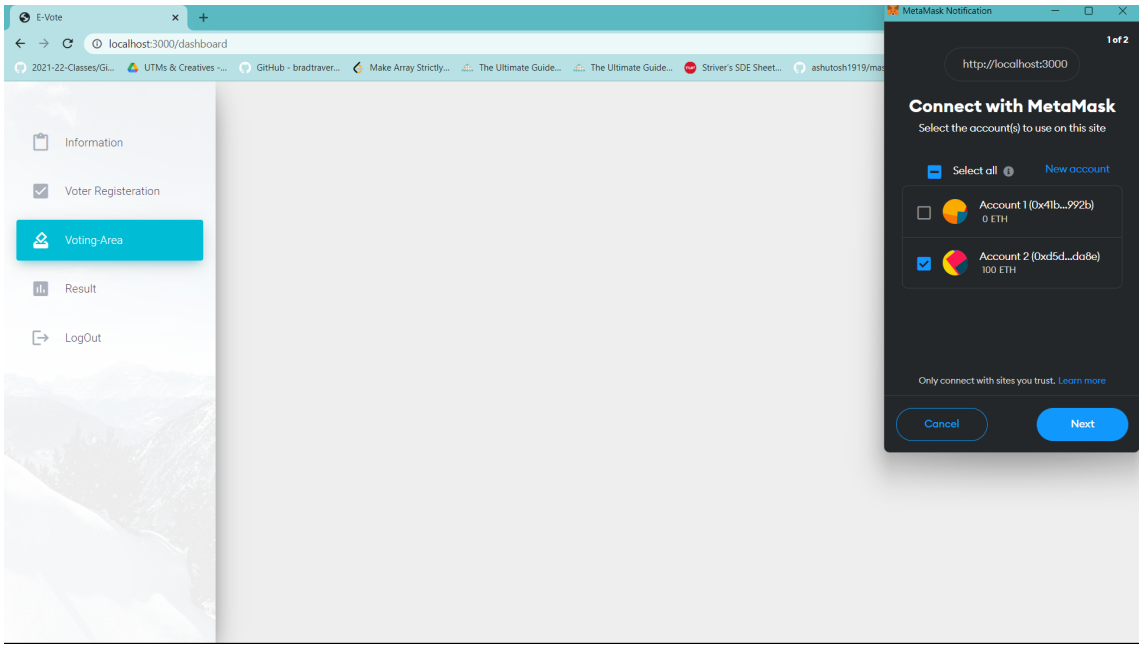
- Next the voter registration tab is there where one can register to vote by entering his aadhar and ethereum wallet address.



- On entering the details, the OTP request page is opened. The OTP is sent to the mail id.

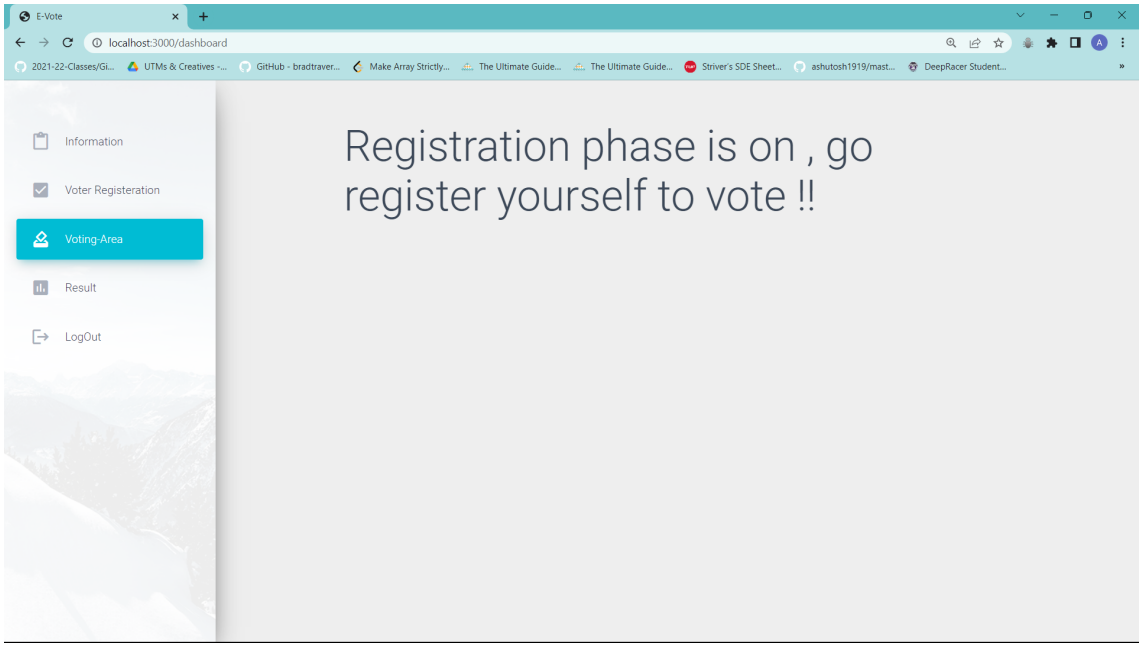


- In the voting area tab, you will be requested to link to your ethereum wallet.

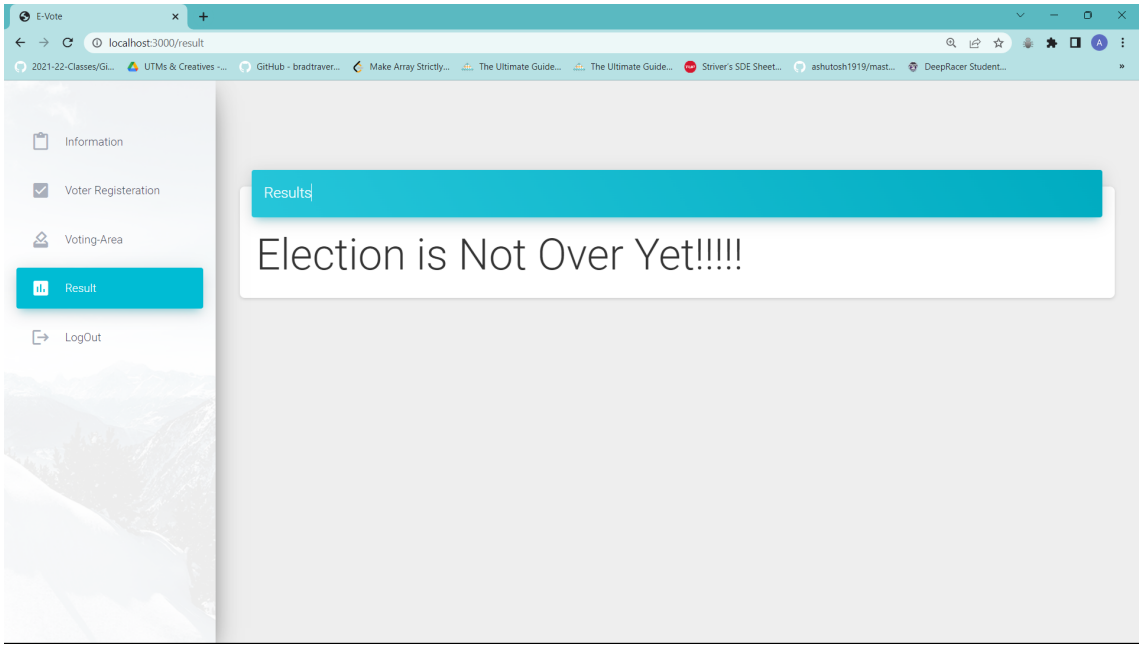


- After connecting to the wallet properly, the following window displays which

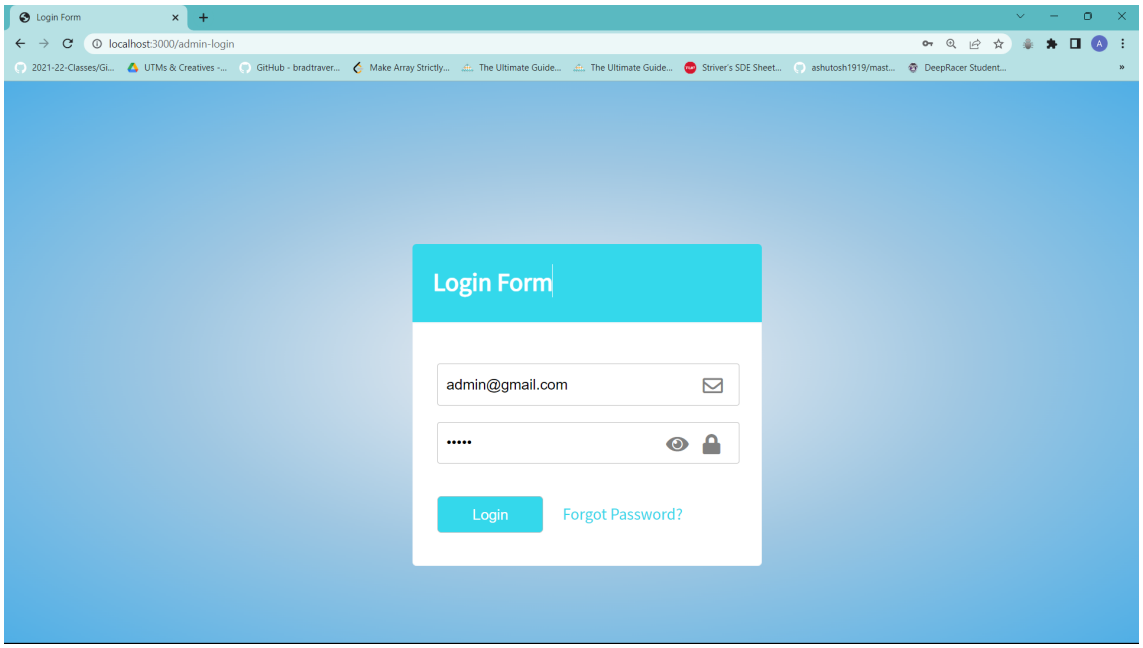
shows that Registration phase is going on and you can't cast vote currently.



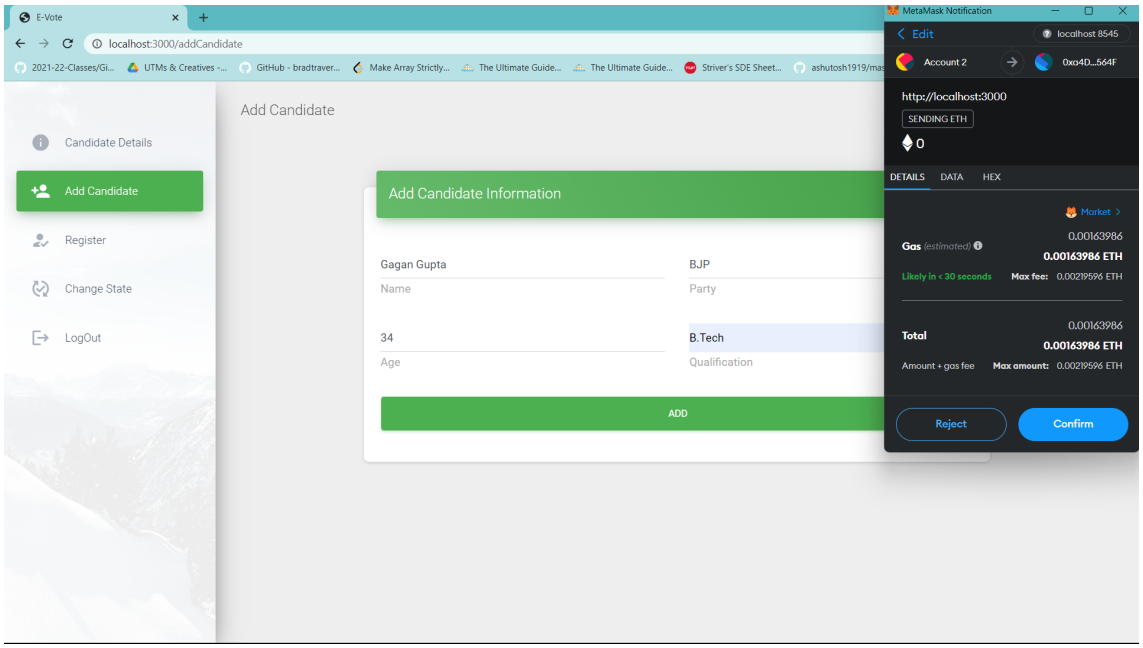
- In the Results tab, since there are no results currently it won't show up anything



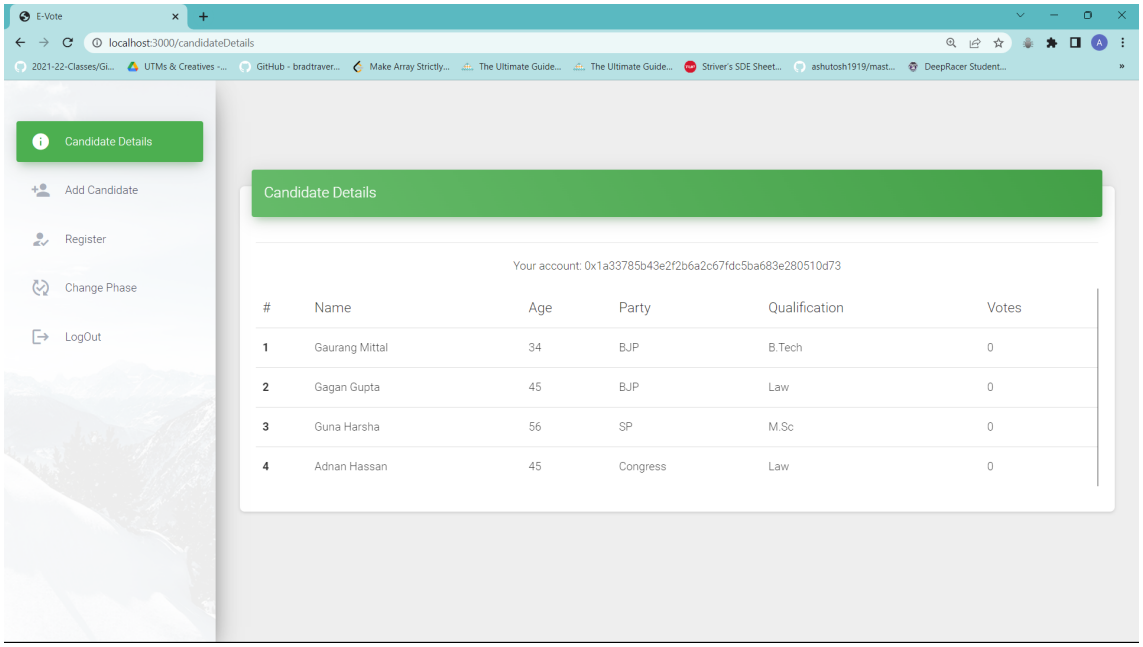
- Log out from the user dashboard and login to the admin panel. Enter the correct credentials and login



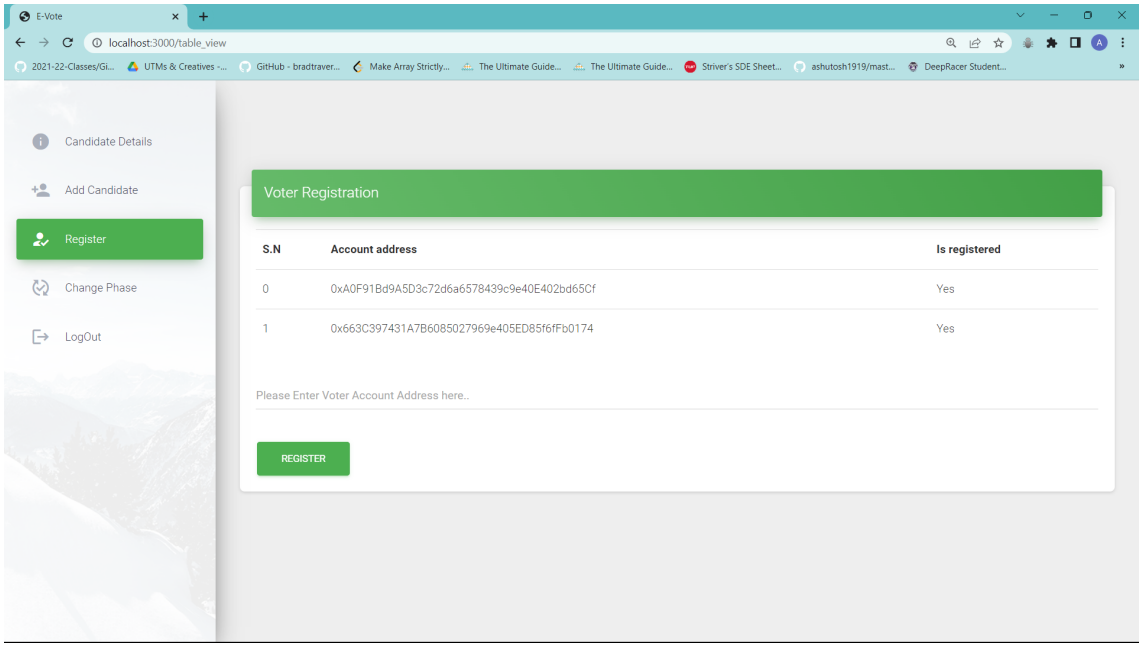
- In the add candidate page, add the credentials of the new candidate who is standing up for the election. The Metamask notification is displayed where it asks to authorize the transaction.



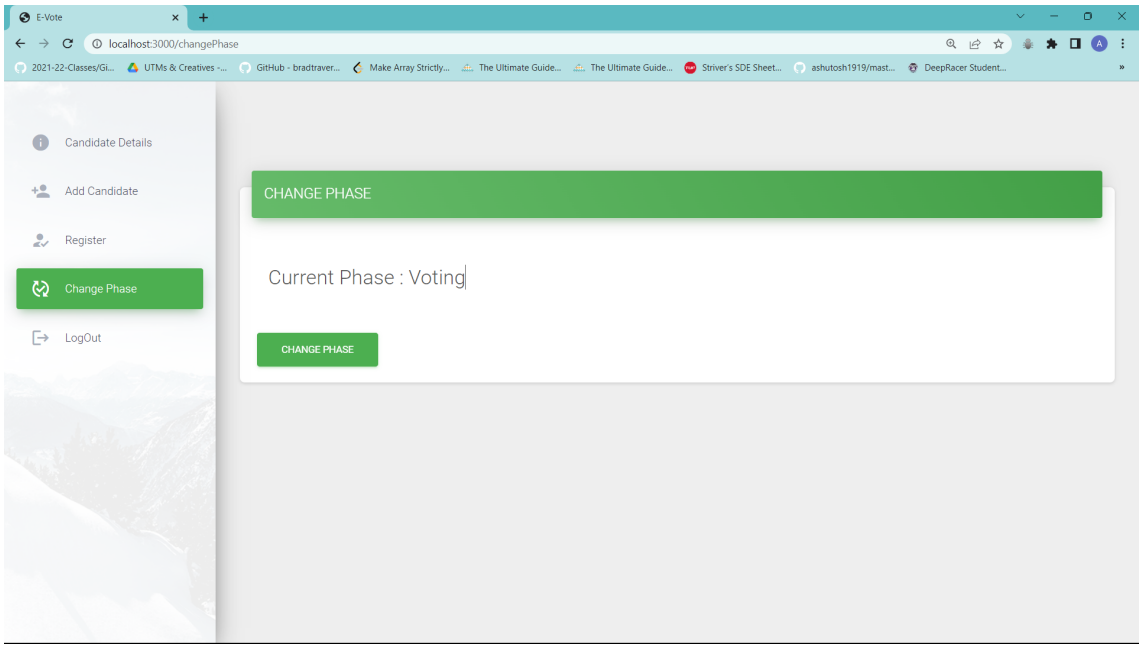
- The transaction creates the new candidate. We can see the candidates standing up for the election in the Candidate Details tab.



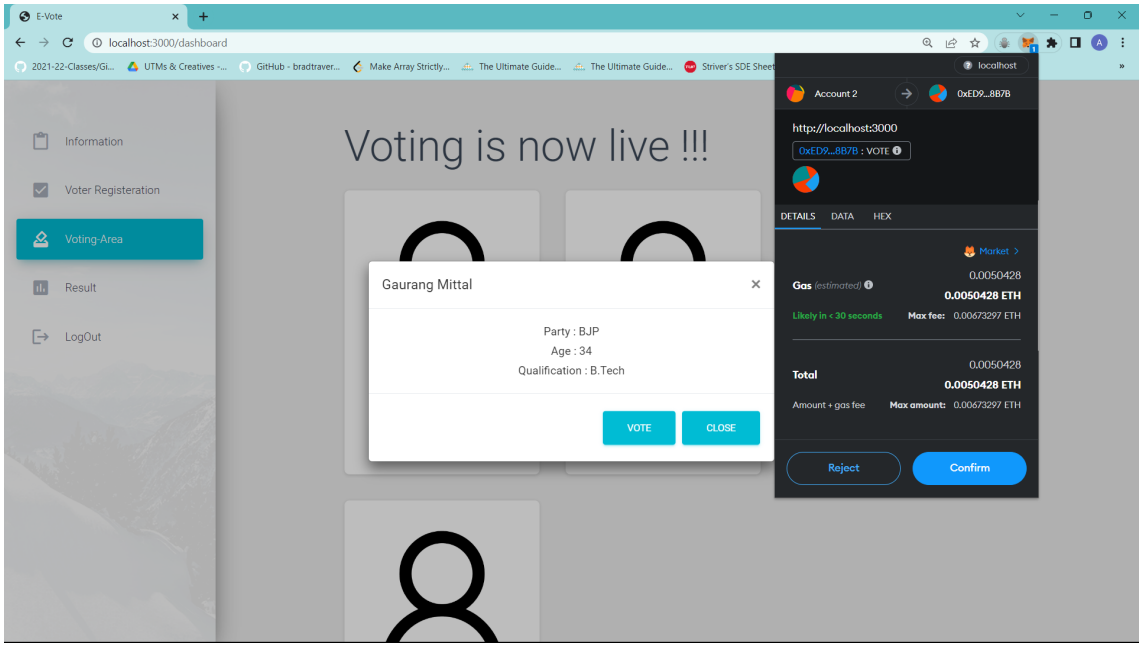
- In the voter registration page,the admin can register the account addresses of the voters who have registered.



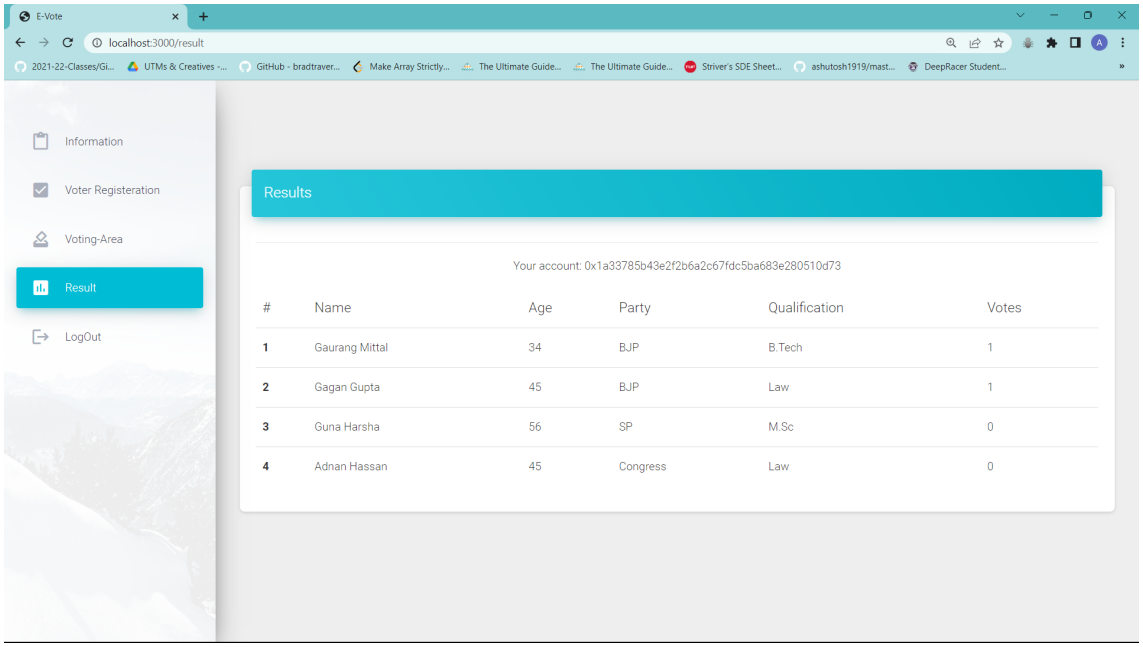
- In the change phase tab, the admin can change the phase of the election. There are three phases : registration, voting and result phase. The initial phase is the registration phase. Whenever the phase is to be changed, the new transaction is done.



- The user can caste his vote now in the voting phase when the registration are closed and voting phase begins



- When the admin changes the phase to the result phase, users can view in the result tab.



Chapter 6

Conclusion

In conclusion, our project BlockBallot, a decentralized voting application using blockchain technology, has the potential to revolutionize the voting system by providing a secure, transparent, and efficient platform for voters to cast their votes. The use of smart contracts and Ethereum blockchain technology ensures that the voting process is tamper-proof and cannot be manipulated by any third party.

Through our experimental and result section, we have shown that the proposed system is reliable, secure, and efficient. The use of Node.js, Express.js, EJS, MySQL, Ganache, Truffle, and Metamask has allowed us to create a robust web application integrated with blockchain technology that is easy to use and accessible to everyone. The use of these technologies not only made the voting process accessible to a larger number of voters but also enhanced the user experience.

This project investigates the potential of using blockchain technology for e-voting systems, making them publicly verifiable and distributed to prevent corruption. The concept of using digital voting systems to improve the electoral process by making it more cost-effective, efficient, and convenient is gaining popularity in modern society. By making the voting process more accessible, it helps to eliminate power imbalances between the elector and the official, and puts pressure on officials to make the process more transparent and trustworthy.

6.1 Future Work

The project is not without its limitations and potential areas for improvement. As with any new technology, there is always room for refinement and enhancement. Future iterations of the BlockBallot project can explore the implementation of various features.

- Currently BlockBallot is authenticating the voter by the OTP through the E-mail which is not the most secure way. We plan to build aadhar based authentication system which is more secure and fraud-free.
- Enhnacing the Graphical User Interface(GUI) of the application.
- Local languages can be included which will play a vital role for people living in rural areas as well as uneducated people.
- Blockchain-based voting systems could be integrated with other technologies, such as artificial intelligence and biometrics, to provide even greater security and accuracy. For example, AI could be used to identify and prevent fraudulent voting behavior, while biometric authentication could be used to ensure that only eligible voters are able to cast their ballots.
- A Candidate's earlier social work and candidate qualification's can be added for a voter to have better choice.
- A complaint system can be included, that allows the people to file complaint against a candidate
- Decentralized blockchain-based voting systems could provide greater security and transparency, but they may face scalability issues as the number of voters and transactions increases. New consensus mechanisms and protocols will need to be developed to address these challenges.

References