“Online Voting System Using Blockchain”

A Project presented to the National University in partial fulfillment of the requirement for the degree of Bachelor of Science (Hon’s) in Computer Science & Engineering.

Submitted By

Sourav Kar
Reg. no.: 17502004923
Session: 2017-18

Submitted To

Md. Imran Hossain
Supervisor & Head
Department of CSE, DIIT

Department of Computer Science &Engineering
Daffodil Institute of IT
Dhanmondi, Dhaka -1205

Submission Date: August 30, 2023

DECLARATION

I hereby declare that the project entitled **“Online Voting System Using Blockchain”** submitted to Department of Computer Science that has been carried under the Supervision of our guide of **Md. Imran Hossain** Lecturer of Computer Science & Engineering, Daffodil Institute of IT as partial fulfillment of requirement Bachelor of Computer Science & Engineering and further certify that this has been previously formed as the award of any degree Diploma of such similar title.

Submitted By

____

Sourav Kar
Reg. no.: 17502004923
Session: 2017-18

ACKNOWLEDGEMENT

I Would like to express my profound gratitude to Almighty Allah. With the blessing of Almighty Allah, I have successfully planned my project.

We would like to thank Prof. Dr. Mohammed Shakhawat Hosaain, Principal, DIIT for encouraging us and giving all the facilities.

We express our sincere thanks to our Md. Imran Hossain, our internal project guide & Head of the Department of Computer Science & Engineering, Daffodil Institute of IT, for this guidance and suggestions during this project work.

We respect and thank all teaching staffs of Department of Computer Science & Engineering who helped us in successfully Completing the project work.

We are thankful to all fortunate enough to get constant encouragement support from almighty, our parents and friends.

ABSTRACT

In any democratic country, Voting is a fundamental right of any citizen that enables them to choose the leaders of tomorrow. Electronic voting or e-voting has fundamental benefits over paper-based systems such as increased efficiency and reduced errors. The blockchain is an emerging, decentralized, and distributed technology with strong cryptographic foundations that promise to improve different aspects of many industries. A blockchain-based voting system that will limit the voting fraud and make the voting process simple, secure and efficient.

Table of Contents

List of Figure

Chapter 01
INTRODUCTION

1.1 Introduction

Electoral integrity is essential not just for popular nations but also for state voter's trust and liability. Political voting styles are pivotal in this respect. Bangladesh is a popular country and has a popular country. As now all Bangladeshi citizen come a part of the growing digital Bangladesh with a digital ID that's National Identity card. The current system, whether electronic or not has proved to be wrong with respect to translucency. The being homemade Voting system consumes further time for Vote Casting. Voter has to stay for vote polling station to bounce for a right seeker. The election officers have to be checked the vote; this voter can bounce in this cell also check voter ID present in choosers list of cells those are information will be present also the voter can bounce in that cell. The voter had to stand in the line to cast his vote. All the work is done in paper ballot so it's veritably hard to detect a particular campaigner, some choosers cast their votes for all campaigners. To overcome of all these problems. we've to apply a web operation, which is helpful for Voting from anywhere. Replacing the traditional system with electronic system using Blockchain fashion has the capability to help implicit frauds that may take place during election

1.2 Problem Definition

The existing manual Voting system consumes more time for Vote Casting. Voters have to wait at the voting polling station to vote for the right candidate. The election officers have to check the voter, this voter can vote in this booth then check voter ID present in voters list of booths. That information will be present then the voter can vote in that booth. The voter had to stand in the queue to cast his vote. All the work is done by paper ballot so it is very hard to locate a particular candidate, some voters cast their votes for all candidates. To overcome all these problems. we have to implement a web application, which is helpful for Voting from anywhere.

1.3 About Project

The objective of the system is a replacement of the traditional system that is in existence. This smart system reduces the time for voting and also the system is reliable, and faster. Database maintained by this system usually contains the Voters information, Candidate information, The final Result of total votes.

1.4 The history of Blockchain

Blockchain can help to implement a system that is immutable, transparent, and efficient and cannot be hacked into. The inability to change or delete information from blocks makes the blockchain the most effective technology for voting systems. Blockchain technology is supported by a distributed network consisting of a variety of interconnected nodes. Each of these nodes have their own copy of the distributed ledger (information) that contains the total history of all transactions the network has processed. There is no centralized system that controls the network. If the majority of the nodes agree, then they accept a transaction. This network permits users to stay anonymous. A basic analysis of the blockchain technology (including sensible contracts) suggests that it is an appropriate basis for e-voting and furthermore, it might have the potential to make e-voting a lot more acceptable and reliable.

Blockchain technology makes e-voting cheaper, easier, and much more secure to implement. It is a considerably new paradigm that can help to form decentralized systems, which assure the data integrity, availability, and fault tolerance. This technology aims to revolutionize the systems. The blockchain systems are formed as decentralized networked systems of computers, which are used for validating and recording the pure online transactions. They also constitute ledgers, where digital data is tied to each other, called the blockchain. The records on the blockchain are essentially immutable.

Chapter 02
LITERATURE SURVEY

2.1 LITERATURE SURVEY

Currently increasing digital technology has helped many people's lives. In contrast to the electoral system, there are many conventional uses of paper in its implementation. The aspect of security and transparency is a threat from still widespread elections with the conventional system. Block chain technology is one of the solutions, because it embraces a decentralized system and the entire database is owned by many users.

There is no doubt that the revolutionary concept of the blockchain, which is the underlying technology behind the famous crypto currency Bitcoin and its successors, is triggering the start of a new era in the Internet and the online services. In this work, we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language.

Block chain was first introduced by Satoshi Nakamoto (a pseudonym), who proposed a peer-to-peer payment system that allows cash transactions through the Internet without relying on trust or the need for a financial institution. Block chain is secure by design, and an example of a system with a high byzantine failure tolerance.

E-voting is a potential solution to the lack of interest in voting amongst the young tech savvy population. For e-voting to become more open, transparent, and independently auditable, a potential solution would be to base it on block chain technology. Block chain technology has a lot of promise; however, in its current state it might not reach its full potential.

Electronic voting has been used in varying forms since the 1970s with fundamental benefits over paper-based systems such as increased efficiency and reduced errors. With the extraordinary growth in the use of block chain technologies, a number of initiatives have been made to explore the feasibility of using block chain to aid an effective solution to e-voting. It presented one such effort which leverages benefits of block chain such as cryptographic foundations and transparency to achieve an effective solution to e-voting. The proposed approach has been implemented with Multichain and in-depth evaluation of the approach highlights its effectiveness with respect to achieving fundamental requirements for an e-voting scheme.

Recent major technical challenges relating to e-voting systems embrace, however not restricted to secure digital identity management. Any potential citizen ought to be registered to the electoral system before the elections. Their data ought to be in a very 5 digitally processable format. Besides, their identity data ought to be unbroken personal in any involving information. Ancient E-voting system could face following problems:

- ❖ Anonymous vote-casting.
- ❖ Individualized ballot processes.
- ❖ Ballot casting verifiability by (and only by) the voter.
- ❖ High initial setup costs.
- ❖ Increasing security problems.
- ❖ Lack of transparency and trust.
- ❖ Voting delays or inefficiencies related to remote/absentee voting.

To mitigate these threats, software mechanisms which promise the following should be deployed and display into two parts

- Prevention of evidence deletion.
- Transparency with privacy.

Using a Blockchain, the most important requirements are satisfied:

- Authentication: Only registered voters will be allowed to vote.
- Anonymity: The system prevents any interaction between the votes casted by the voters and their identities.
- Accuracy: Votes once cast are permanently recorded and cannot be modified or changed under any circumstances.
- Verifiability: The system will be verifiable such that the number of votes is accounted for.

As technology advances, many countries have now opted for electronic voting systems. Any voting system must follow principles of transparency and impartiality in order to achieve fairness; the electronic voting process must also be protected against cyberattacks or denial-of-service attacks (DDOS) because such attacks may affect the processing time in voting procedures and even hinder the fairness in voting. This study establishes a network security mechanism for voting systems based on blockchain technology. The blockchain mechanism employs a distributed architecture that can prevent system shutdown resulting from malicious cyber-attacks; additionally, any user in the blockchain can authenticate data integrity, which satisfies requirements of transparency and impartiality in voting systems.

Chapter 03
Project Analysis

3.1 EXISTING SYSTEM

The Existing System of Election is running manually. The Voter has to Visit to Booths to Vote a Candidate so there is wastage of Time. Due to this many people don't go out to cast their vote which is one of the most important and Worrying factors. In democracy Each and every vote is important. This Traditional system can be replaced by a new online system which will limit the voting frauds and make the voting as well as counting more efficient and transparent.

3.2 Proposed system

The current voting system requires some improvement in it because of the issues mentioned above. This can be achieved by replacing the existing system with a new system which will limit the voting frauds and make the voting as well as counting more efficient.

- ❖ The Online Election System would have user registration, user login and admin login.
- ❖ This Online Voting System will manage the Voter's information by which a voter can login and use his voting rights.
- ❖ At the time of registration voters will be asked for this: Full name, age, National Identity card no, mobile no. email id and after being verified will be given the access.
- ❖ At the time of requesting a vote, the voter will be asked to enter his National Identity id. Then the voter will be authenticated, and he can vote for one of the candidates from the list. Voters can vote for a Candidate only once per Election.
- ❖ The software system allows the user to login in to their profiles and upload all their details including their previous milestone onto the system. The admin can check each Candidate's details.
- ❖ The software system also allows Voters to view a list of Candidates in their area. The admin has overall rights over the system and can moderate and delete any details not pertaining to Election Rules.

3.3 Benefits of E-voting system over the current system

Benefits of E-voting system over the current system:

- Increasing the level of participation: The Internet voting system tends to maximize user participation, by allowing them to vote from anywhere and from any device that has an internet connection.
- Security: By considering the importance of the e-voting system is implemented using “Blockchain”.
- Efficiency: The reduction in organizational and implementation costs significantly increases the efficiency of election management compared to traditional paper voting, for example.
- Precision: The electronic vote eliminates errors in manual count, which brings with it an accurate and quick publication of results, with receipt of vote for each vote cast.

Chapter 04
System Requirements

4.1 Introduction

In this chapter, we will discuss the tools we have used to complete this project. We have divided the tools into several sections. Such as hardware, software, platform, etc.

4.2 Hardware Requirements

Hardware requirements refer to the physical components and specifications that are needed to run a particular software system. The specific hardware requirements can vary depending on the complexity and size of the software, as well as the operating system and other software that it will be running on.

It is important to check the specific hardware requirements of the software that you want to install or run, as these may vary depending on the software's complexity, version, and specific features.

4.3 Desktop or Laptop Requirements

- ❖ Processor: Core i3
- ❖ RAM: 4GB
- ❖ Hard Disk: 500 Gb
- ❖ Speed: 1.1GHz

4.4 Software Requirements

Software requirements refer to the functional and non-functional specifications that a software system should meet to satisfy the needs of its users. Here are some key elements that should be included in software requirements:

Functional requirements: These requirements define the specific tasks and functions that the software should perform, such as data input and processing, calculations, and report generation.

Non-functional requirements: These requirements define the characteristics of the software that are not directly related to its functionality, such as performance, reliability, usability, and security.

User requirements: These requirements specify the needs of the users who will be interacting with the software, including their skill level, expectations, and preferences.

Technical requirements: These requirements define the technical specifications that the software must meet, such as operating system compatibility, hardware requirements, and programming languages and tools.

Maintenance requirements: These requirements specify the maintenance and support needs of the software, including software updates, bug fixes, and user support.

4.5 Language and Framework

- Operating System: Windows
- Type: Web Application.
- Front-End: HTML5 and CSS3, JS
- Back-End: Solidity, Node.js, EJS, Express.js
- Framework: Truffle
- Extensions: Meta Mask
- Server: (cross platform, Apache)
- Database: MYSQL
- Supporting Tools: Visual studio code, Terminal, NPM, Node-mailer, Ganache
- Web Browser: Google Chrome, Firefox, Microsoft Edge, or any other HTML5 supported.

Chapter 05
Tools and Technologies Used

5.1 Ganache

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle; enabling you to develop, deploy, and test your daps in a safe and deterministic environment.

Ganache comes in two flavors: a GUI and CLI. Ganache UI is a desktop application supporting Ethereum and File coin technology. Our more robust command-line tool, ganache, is available for Ethereum development. It offers:

- console.log in Solidity
- Zero-config Main net and test net forking
- Fork any Ethereum network without waiting to sync
- Ethereum JSON-RPC support
- Snapshot/revert state
- Mine blocks instantly, on demand, or at an interval
- Fast-forward time
- Impersonate any account (no private keys required!)
- Listens for JSON-RPC 2.0 requests over HTTP/WebSocket
- Programmatic use in Node.js
- Pending Transactions

5.2 Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, and Go) and run times (such as .NET and Unity). Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. Instead of a project system, it allows users to open one or more directories, which can then be saved in the workspace for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which uses proprietary. In solution files and project-specific project files. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, add code linters, use the Language Server Protocol and connect to additional services. Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character for Windows/Linux, and the programming language of the active document.

5.3 Node mailer

Node mailer is a module for Node.js applications to allow easy as cake email sending. The project got started back in 2010 when there was no sane option to send email messages, today it is the solution most Node.js users turn to by default.

- ❖ A single module with zero dependencies – code is easily auditable, as there are no dark corners
- ❖ Heavy focus on security, no-one likes RCE vulnerabilities
- ❖ Unicode support to use any characters, including emoji
- ❖ Windows support – you can install it with npm on Windows just like any other module, there are no compiled dependencies. Use it hassle free from Azure or from your Windows box
- ❖ Use HTML content, as well as plain text alternative
- ❖ Add Attachments to messages
- ❖ Embedded image attachments for HTML content – your design does not get blocked
- ❖ Sign messages with DKIM
- ❖ Custom Plugin support for manipulating messages
- ❖ Sane OAuth2 authentication

- Proxies for SMTP connections

5.4 Meta Mask

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. MetaMask is developed by Consent Sys Software Inc., a blockchain software company focusing on Ethereum-based tools and infrastructure.

MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser. Websites or other decentralized applications are able to connect, authenticate, and/or integrate other smart contract functionality with a user's MetaMask wallet (and any other similar blockchain wallet browser extensions) via JavaScript code that allows the website to send action prompts, signature requests, or transaction requests to the user through MetaMask as an intermediary.

The application includes an integrated service for exchanging Ethereum tokens by aggregating several decentralized exchanges (DEXs) to find the best exchange rate. This feature, branded as MetaMask Swaps, charges a service fee of 0.875% of the transaction amount. As of November 2021, MetaMask's browser extension had over 21 million monthly active users, according to Bloomberg.

5.4 Truffle

Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. Truffle is widely considered the most popular tool for blockchain application development with over 1.5 million lifetime downloads. Truffle supports developers across the full lifecycle of their projects, whether they are looking to build on Ethereum, Hyperledger, Quorum, or one of an ever-growing list of other supported platforms. Paired with Ganache, a personal blockchain, and Drizzle, a front-end App development kit, the full Truffle suite of tools promises to be an end-to-end App development platform.

- ❖ Built-in smart contract compilation, linking, deployment and binary management. • Automated contract testing for rapid development.
- ❖ Scriptable, extensible deployment & migrations framework.
- ❖ Network management for deploying to any number of public & private networks. • Package management with EthPM & NPM, using the ERC190 standard.
- ❖ Interactive console for direct contract communication.
- ❖ Configurable build pipeline with support for tight integration.
- ❖ External script runner that executes scripts within a Truffle environment.

5.5 Solidity

Solidity is an object-oriented programming language created specifically by the Ethereum Network team for constructing and designing smart contracts on Blockchain platforms.

- ❖ It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.
- ❖ It acts as a tool for creating machine-level code and compiling it on the Ethereum Virtual Machine (EVM).
- ❖ It has a lot of similarities with C and C++ and is pretty simple to learn and understand. For example, a “main” in C is equivalent to a “contract” in Solidity.

Like other programming languages, Solidity programming also has variables, functions, classes, arithmetic operations, string manipulation, and many other concepts.

Some key features of solidity are listed below:

- ❖ Solidity is a high-level programming language designed for implementing smart contracts.
- ❖ It is a statically-typed object-oriented(contract-oriented) language.

- Solidity is highly influenced by Python, C++, and JavaScript which runs on the Ethereum Virtual Machine (EVM).
- Solidity supports complex user-defined programming, libraries and inheritance.
- Solidity is the primary language for blockchains running platforms.
- Solidity can be used to create contracts like voting, blind auctions, crowdfunding, multi signature wallets, etc.

5.6 Introduction to JS (JavaScript)

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defines a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

5.7 Introduction to CSS (Cascading Style Sheet)

CSS stands for Cascading Style Sheets. CSS describe hot HTML elements are to be displayed on the screen, paper or in other media CSS saves a lot of work It can control the layout of multiple web pages all at once External style sheets are stored in CSS files.CSS are utilized to organize the format of web pages CSS helps web engineers make a uniform look over a few pages of a website. Once the style in CSS, it can be utilized by any page that references the CSS record. CSS can be a language with regard to specifying exactly how documents tend to be presented in order to users - how they are usually styled, organized, etc. The document is generally a text data file structured by using a markup vocabulary - HTML page is the most typical markup terminology, but we will even come across various other markup different languages such as SVG or XML.

5.8 Introduction to HTML (Hyper Text Markup Language)

HTML (Hypertext Markup Language) is a computer language devised to allow website creation. Anyone else connected to the Internet can then view these websites. It is relatively easy to learn, with the basics being accessible to most people in one sitting: and quite powerful in what it allows you to create. As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>.** Except for a few tags, most of the tags have their corresponding closing tags. For example**, <html>** has its closing tag **</html>** and <body> tag has its closing tag **</body>** tag, etc. Using HTML5 helps to eliminate most <div> tags and replace them with semantic elements. Designers can now use cleaner and neater code. Make a more detailed understanding of the structure of a page with the help of HTML5. Make a more thorough understanding of the structure of a page with the help of HTML5. With its new features standards, HTML5 makes it easier to create front-end applications such as drag-and-drop tools, wikis, discussing boards, and other useful elements. HTML5 provides a smarter solution to specify the files that the browser should cache; the page can be loaded correctly even when offline.

Chapter 06
MODULES OF PROPOSED SYSTEM

6.1 ADMINISTRATIVE MODULE

Online Voting is a voting system by which any Voter can use his\her voting rights from anywhere in Bangladesh.

Admin- The admin module is divided into 5 components-

- ❖ Dashboard- It will contain various charts to display information such as number of parties, number of voters etc.
- ❖ Add Candidate - In this feature of admin, he can add candidates who are standing in the election. After a candidate is added it will be displayed on the user side.
- ❖ Create Election- This feature of admin will allow him to create elections. A user can cast his vote only after the election is created by the admin. A user can cast a vote between the start date and end date.
- ❖ Election Details- In this section admin can update election details such as start date, end date etc.
- ❖ Candidate Details- In candidate details all the candidates added by admin will be displayed. Admin can update the candidate details if incase a wrong entry is done.

6. 2 USER/VOTER MODULE

User- The user module is divided into 4 components

- ❖ Dashboard- The user dashboard contains information about parties and their candidates. A user can see all the information about a candidate.
- ❖ Voter Register- In this section first user will have to register himself only then he will be able to cast his vote.
- ❖ Voting Area- After a user is registered, then only he will be directed to this page and then he can cast his vote.
- ❖ Results- In this component the user will be able to see the results of the election.

6.3 Nominee Candidate Module

The Nominee details will be updated by the admin for the post of board of director and manager. The candidate will submit their own details and the admin maintains all of the background details of the particular nominee and uploads their information in the correct procedure. In order to, the user or voter can view the nominee details.

Chapter 07
DESIGN and Implementation

7.1 Data Flow Diagram

The data flow diagram (DFD) is a graphical tool used for expressing system requirements in a graphical form. The DFD is also known as the “bubble chart” as the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. Thus, DFD can be stated as the starting point of the design phase that functionality decomposes the requirements specification down to the lowest level of details. The DFD consists of a series of bubbles joined by lines. The bubble represents data transformation and the lines represent the data flows in the system.

- ❖ Arrow: An arrow identifies the data flow in motion. It is a pipeline through which information flows like the rectangle in the flowchart.
- ❖ Circle: A circle stands for process that converts data into information
- ❖ Open End Box: An open-ended box represents a data store, data at rest or a temporary repository of data.
- ❖ Squares: A square defines a source or destination of a system.

7.2 Activity diagram for User

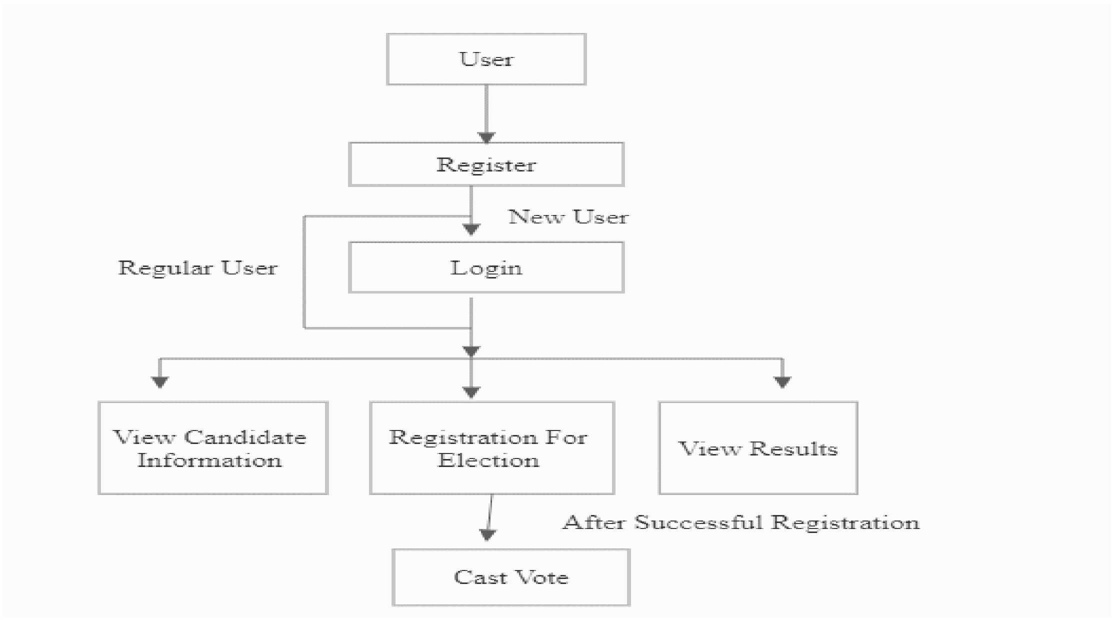

User
Register
New User
Regular User
Login
View Candidate Information
Registration For Election
View Results
After Successful Registration
Cast Vote

Fig: 7.2 Activity diagram for User

7.3 Activity diagram for Admin

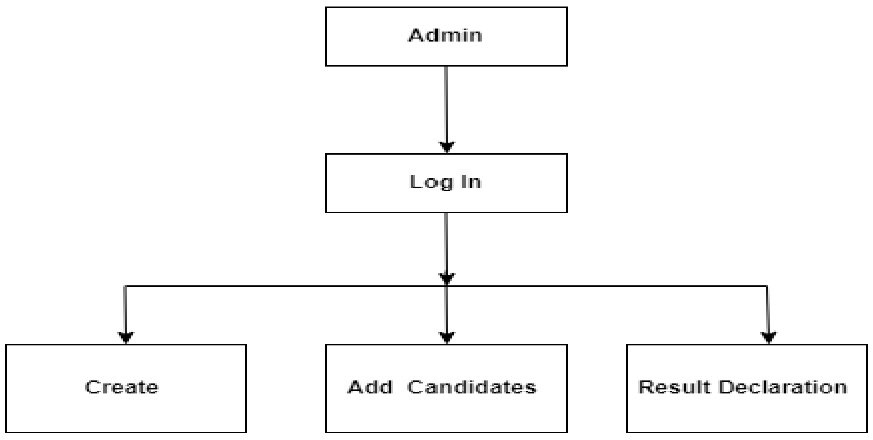


Fig: 7.3 Activity diagram for Admin

7.4 Research methodology

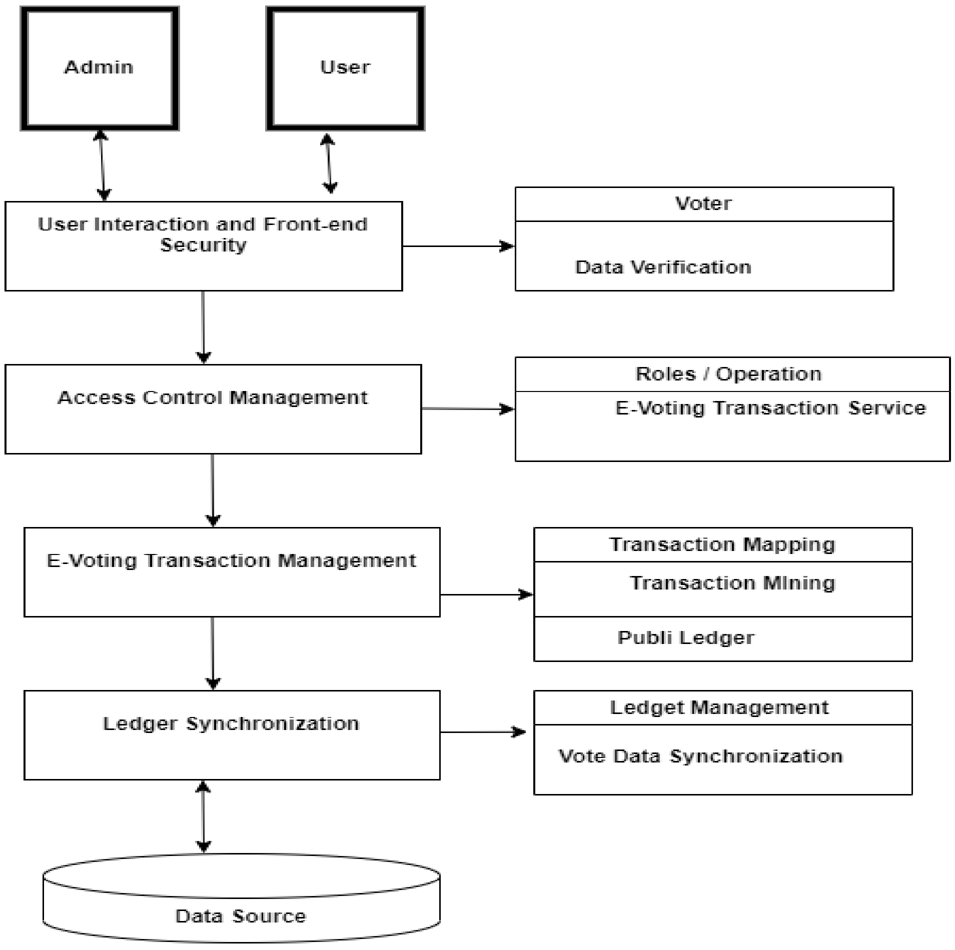

Admin
User
User Interaction and Front-end Security
Voter
Data Verification
Access Control Management
Roles / Operation
E-Voting Transaction Service
E-Voting Transaction Management
Transaction Mapping
Transaction MIning
Publi Ledger
Ledger Synchronization
Ledget Management
Vote Data Synchronization
Data Source

Fig: 7.4 Research methodology

7.5 UI of Website

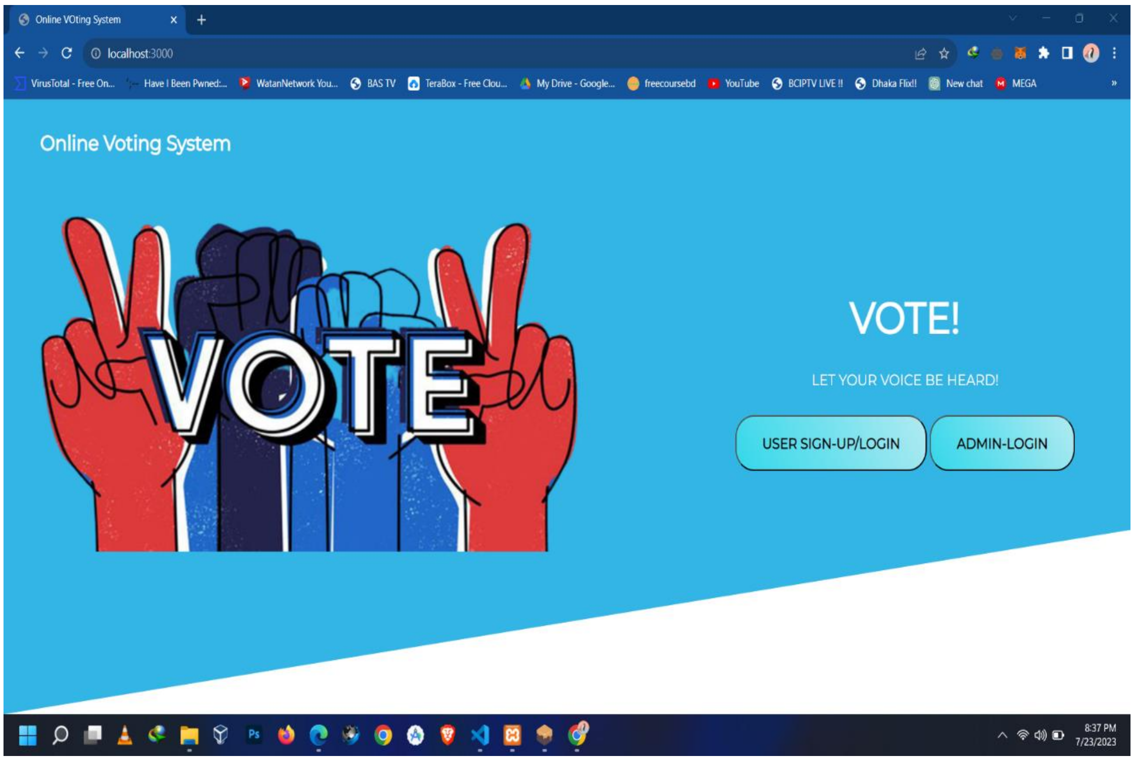


Fig 7.5.A: Homepage

The homepage consists of two options –

- ❖ First option is for admin login
- ❖ Second option is for user signup/login

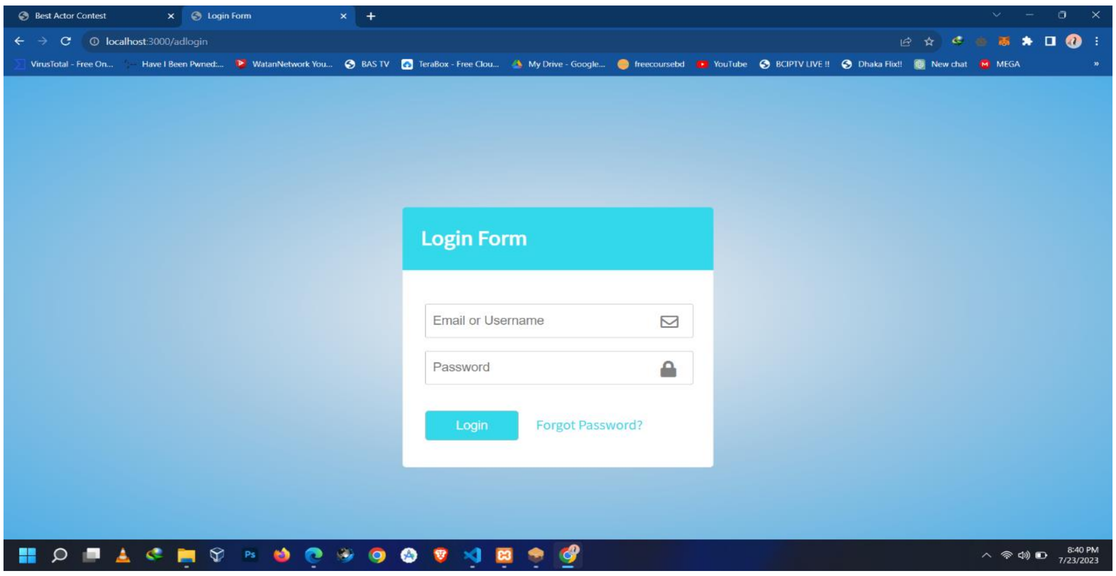
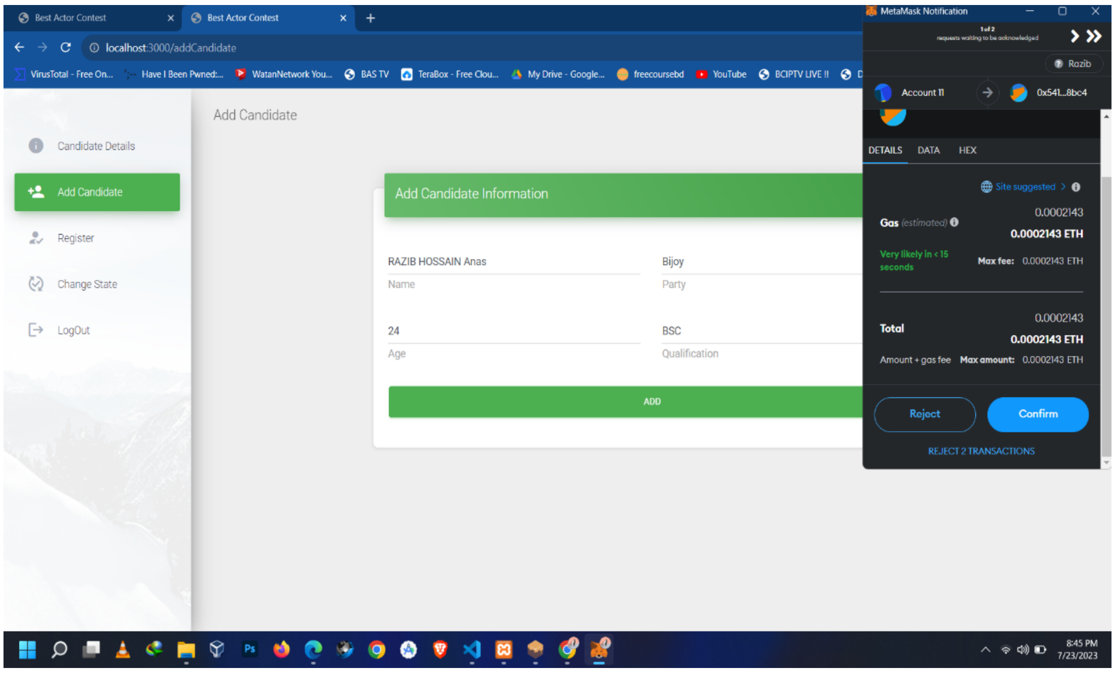


Fig 7.5.B: Admin login

- ❖ This is the login page for admin.
- ❖ After admin is logged in he is directed to the dashboard



Add Candidate

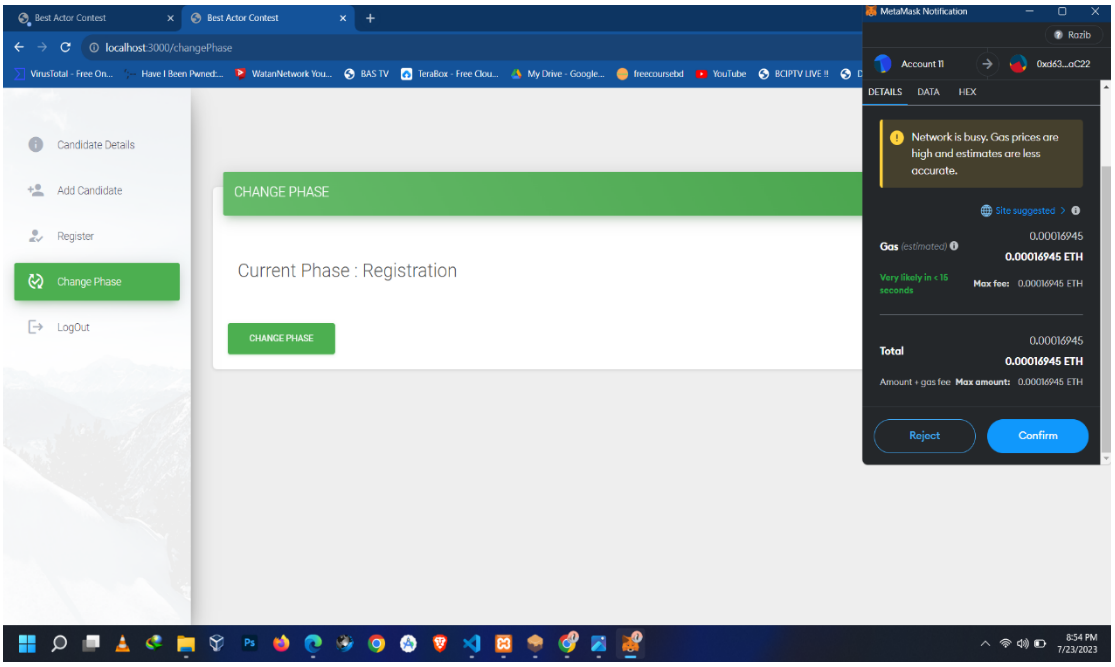
Add Candidate Information

RAZIB HOSSAIN Anas — Name
Bijoy — Party
24 — Age
BSC — Qualification

Account 11 → 0x541...8bc4
Gas (estimated) 0.0002143 — 0.0002143 ETH
Very likely in < 15 seconds — Max fee: 0.0002143 ETH
Total 0.0002143 — 0.0002143 ETH
Amount + gas fee — Max amount: 0.0002143 ETH

Fig 7.5.C: Add Candidates

- ❖ In this feature of admin, he can add candidates who are standing in the election.
- ❖ After a candidate is added it will be displayed on the user side.



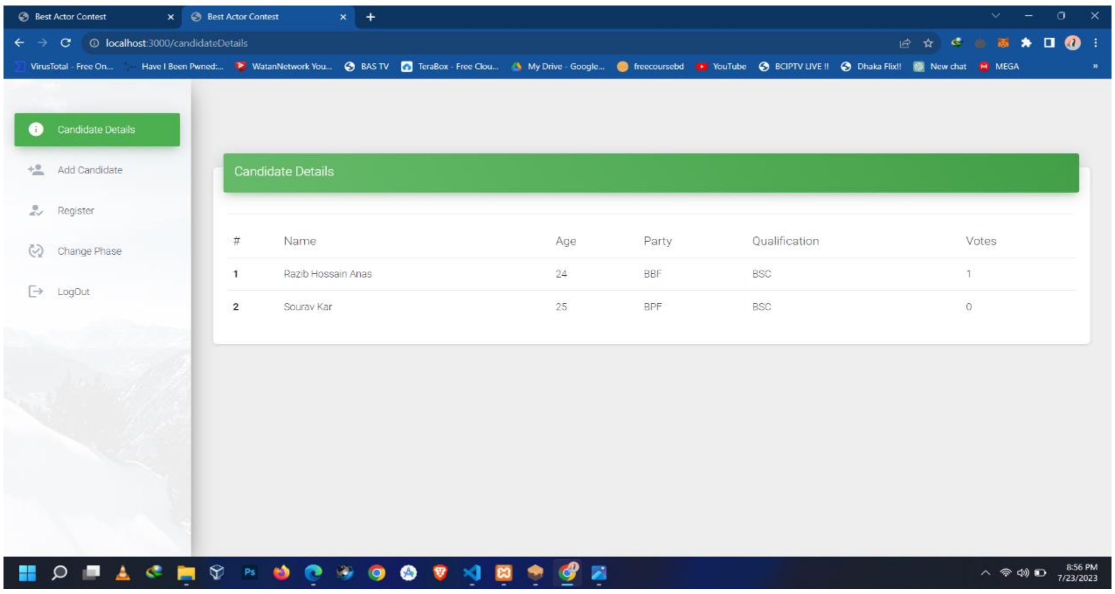
CHANGE PHASE

Current Phase : Registration

Account 11 → 0xd63...aC22
Network is busy. Gas prices are high and estimates are less accurate.
Gas (estimated) 0.00016945 — 0.00016945 ETH
Very likely in < 15 seconds — Max fee: 0.00016945 ETH
Total 0.00016945 — 0.00016945 ETH
Amount + gas fee — Max amount: 0.00016945 ETH

Fig 7.5.D: Election phase

❖ This feature of admin will allow him to create elections.
❖ A user can cast his vote only after the election is created by the admin.



#	Name	Age	Party	Qualification	Votes
1	Razib Hossain Anas	24	BBF	BSC	1
2	Sourav Kar	25	BPF	BSC	0

Fig 7.5.E: Candidates Detailed

In candidate details all the candidates added by admin will be displayed. Admin can update the candidate details if incase a wrong entry is done.

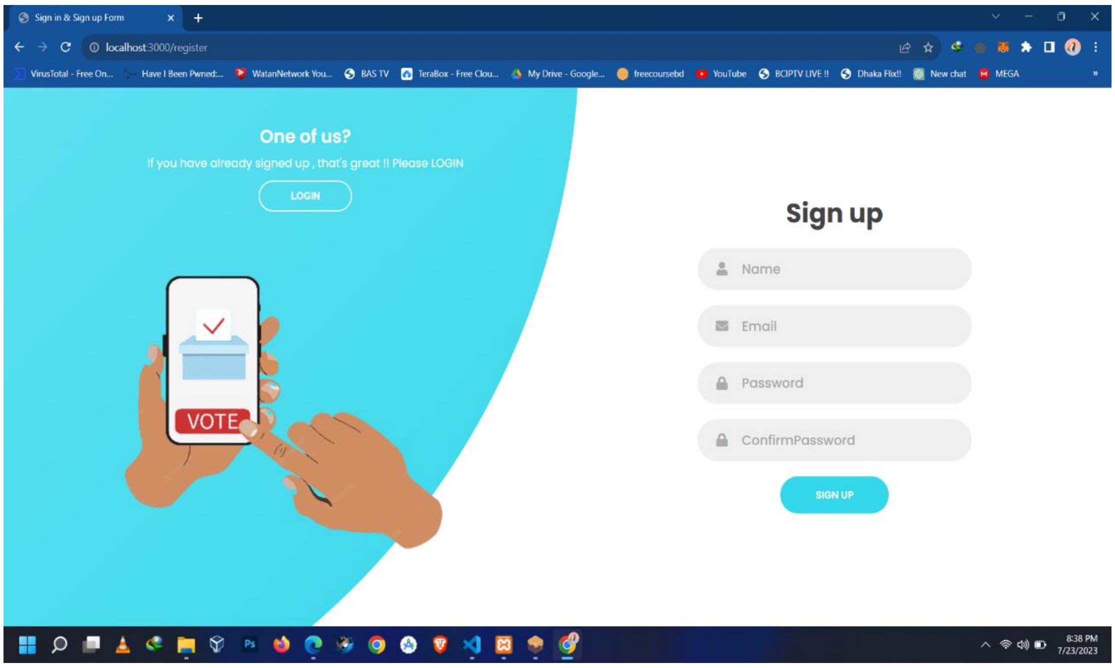


Fig 7.5.F: User SignUp

❖ Users will have to sign up before login.
❖ This is the signup page for the user.

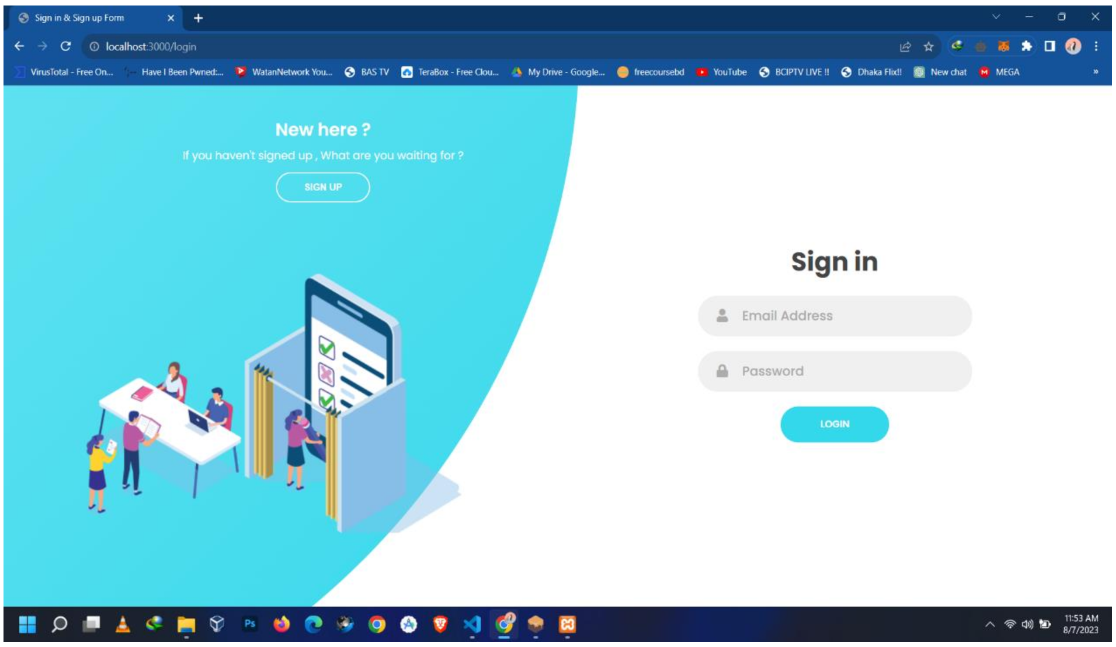
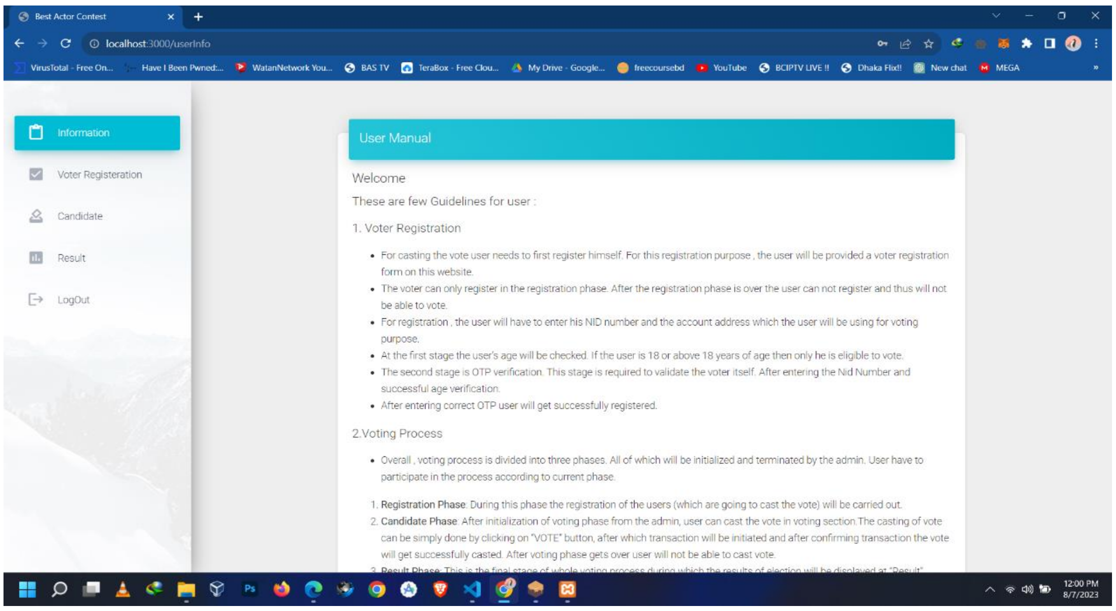


Fig 7.5.G: User Login

❖ If user chooses the user sign in option on homepage he will be directed to this page.
❖ After login he will be directed to the user dashboard.



Welcome

These are few Guidelines for user :

1. Voter Registration

- For casting the vote user needs to first register himself. For this registration purpose , the user will be provided a voter registration form on this website.
- The voter can only register in the registration phase. After the registration phase is over the user can not register and thus will not be able to vote.
- For registration , the user will have to enter his NID number and the account address which the user will be using for voting purpose.
- At the first stage the user's age will be checked. If the user is 18 or above 18 years of age then only he is eligible to vote.
- The second stage is OTP verification. This stage is required to validate the voter itself. After entering the Nid Number and successful age verification.
- After entering correct OTP user will get successfully registered.

2.Voting Process

- Overall , voting process is divided into three phases. All of which will be initialized and terminated by the admin. User have to participate in the process according to current phase.

1. **Registration Phase**: During this phase the registration of the users (which are going to cast the vote) will be carried out.
2. **Candidate Phase**: After initialization of voting phase from the admin, user can cast the vote in voting section.The casting of vote can be simply done by clicking on "VOTE" button, after which transaction will be initiated and after confirming transaction the vote will get successfully casted. After voting phase gets over user will not be able to cast vote.
3. **Result Phase**: This is the final stage of whole voting process during which the results of election will be displayed at "Result"

Fig 7.5.H: User Guidelines

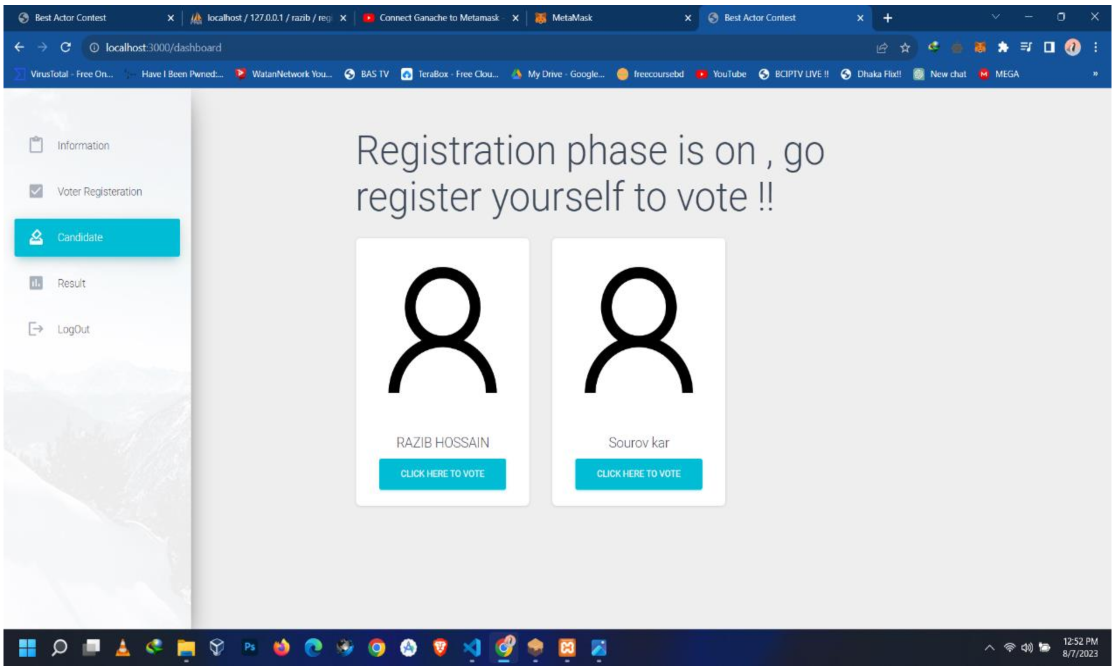
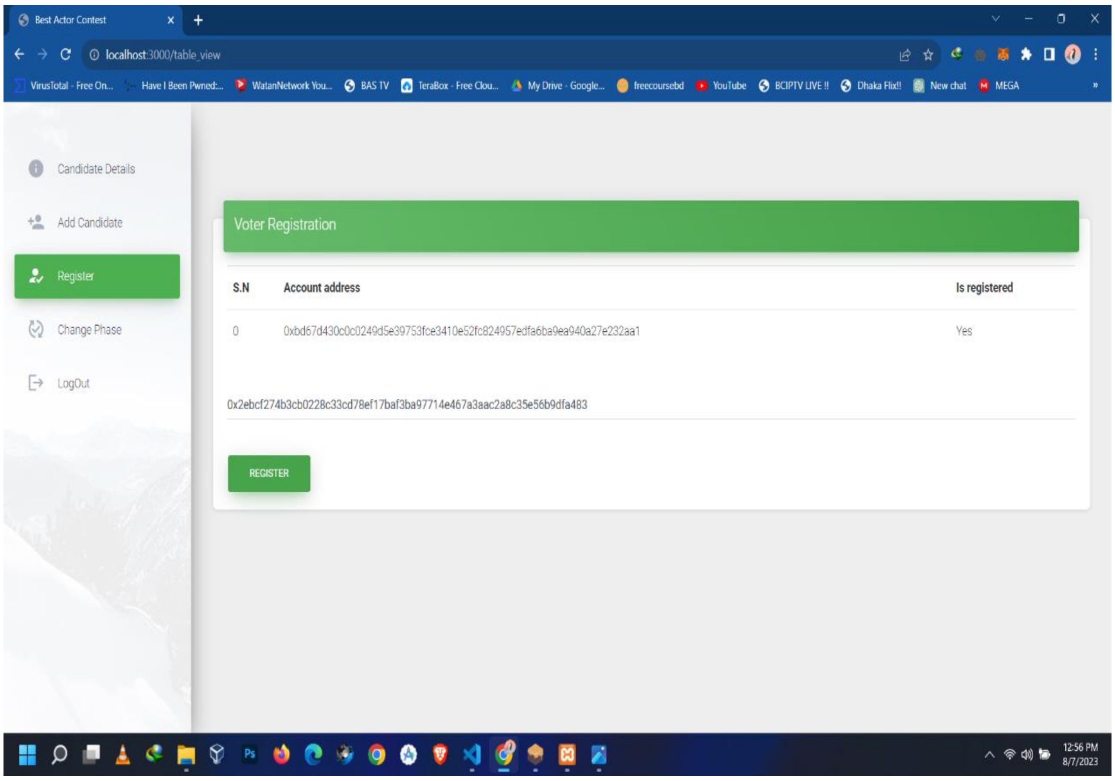


Fig 7.5.I: Candidates

- ❖ The user dashboard contains information about parties and their candidates.
- ❖ A user can see all the information about a candidate.

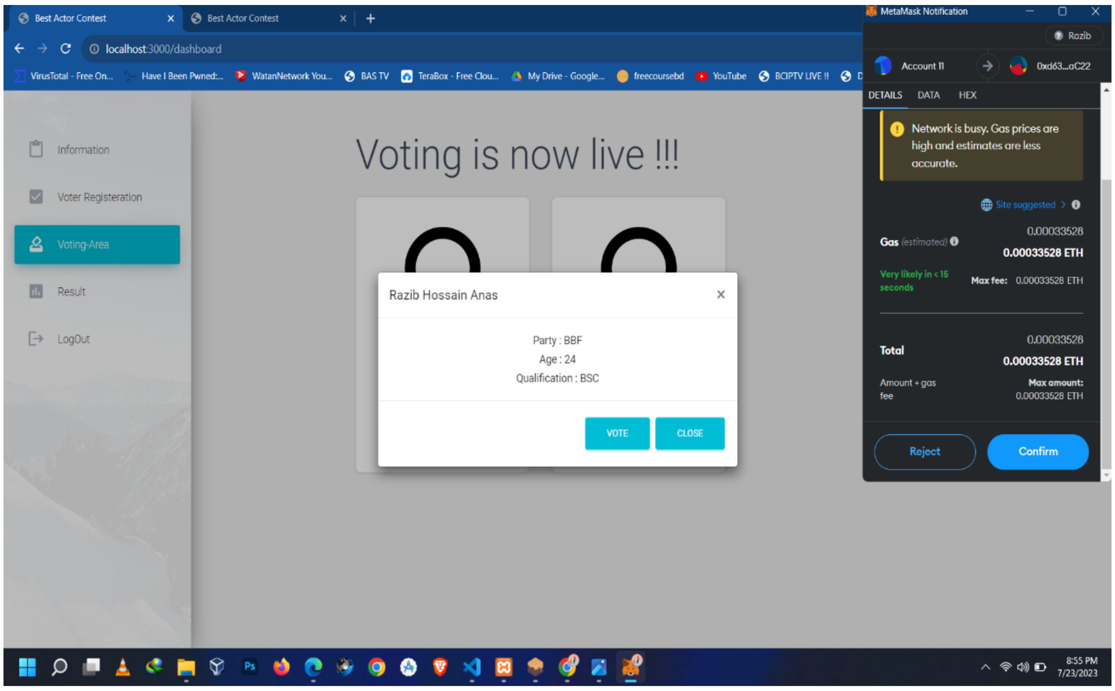


S.N	Account address	Is registered
0	0xbd67d430c0c0249d5e39753fce3410e52fc824957edfa6ba9ea940a27e232aa1	Yes

0x2ebcf274b3cb0228c33cd78ef17baf3ba97714e467a3aac2a8c35e56b9dfa483

Fig 7.5.J: Voter Registration

In this section the first user will have to register himself only then he will be able to cast his vote.



Voting is now live !!!

Razib Hossain Anas

Party : BBF
Age : 24
Qualification : BSC

Network is busy. Gas prices are high and estimates are less accurate.

Gas (estimated) 0.00033528
0.00033528 ETH
Very likely in < 15 seconds
Max fee: 0.00033528 ETH

Total 0.00033528
0.00033528 ETH
Amount + gas fee
Max amount: 0.00033528 ETH

8:55 PM
7/23/2023

Fig 7.5.K: Voting

After the user is registered, then only he will be directed to this page and then he can cast his vote.

Chapter 08
Conclusion

8 .1 CONCLUSION

In this project, we introduced a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient elections while guaranteeing voters privacy. Blockchain technology offers a new possibility to overcome the limitations and adoption barriers of electronic voting systems which ensures the election security and integrity and lays the ground for transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain.

In the future to make the voting process more secure and to correctly identify the person who is voting we can use ML and AI concepts. Using these concepts, we can verify whether the person voting is the same as the person who has registered during the registration process.

REFERENCE

[1] Aayushi Gupta1, Jyotirmay Patel2, Mansi Gupta1, Harshit Gupta1 (2017); *Issues and Effectiveness of Blockchain Technology on Digital Voting; International Journal of Engineering and Manufacturing Science.* ISSN 2249-3115 Vol. 7, No. 1 (2017). [Online]. Available: https://www.ripublication.com/ijems_spl/ijemsv7n1_04.pdf

**[2] Pavel Tarasov and Hitesh Tewari (2017)*; the Future of E-Voting; IADIS International Journal on Computer Science and Information Systems* Vol.12, No. 2, pp. 148- 165 I. [Online]. Available: https://www.researchgate.net/publication/321803764_THE_FUTURE _OF_EVOTING

[3] Coursera. [Online]. Available: https://www.coursera.org/specializations/blockchaina

[4] Edureka (How Blockchain Works) - Simply Explained. [Online]. Available: https://youtu.be/9qfxLo1rt1Q?list=PL9ooVrP1hQOFJblZm3OdcVVH 6Z8V7HP1

[5] Introduction to Blockchain by NPTEL IITM. [Online] Available: https://youtu.be/mzPoUjQC4WU

[6] Blockchain in e-voting. [Online]. Available: https://www.youtube.com/watch?v=d0iLN8LDJ8g&feature=youtu.be

[7] Umut Can Çabuk1, Eylül Adıgüzel2, Enis Karaarslan2 (2018); *A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems; International Journal of Advanced Research in Computer and Communication Engineering.* [Online]. Available: https://www.researchgate.net/publication/324038634_A_Survey_on_F easibility_and_Suitability_of_Blockchain_Techniques_for_the_E - Voting_Systems

Source code

Admin Login

```
var express = require('express');

var router = express.Router();
var db=require('../database');
var app = express();
app.use(express.static('public'))
app.use('/css',express.static(__dirname + 'public/css'))
/* GET users listing. */
router.get('/adlogin', function(req, res, next) {
  res.render('admin_login.ejs');
});


router.post('/adlogin', function(req, res){
    var emailAddress = req.body.email_address;
    var password = req.body.password;

    var sql='SELECT * FROM registration WHERE email_address =?
AND password =?';
    db.query(sql, [emailAddress, password], function (err,
data, fields) {
        if(err) throw err
        if(data.length>0){
            req.session.loggedinUser= true;
            req.session.emailAddress= emailAddress;
            res.redirect('/addCandidate');
        }else{
            res.render('admin_login.ejs',{alertMsg:"Your Email
Address or password is wrong"});
        }
    })

})

module.exports = router;
```

Dashboard

```
var express = require('express');
// var auth = require('auth');
var router = express.Router();
/* GET users listing. */
router.get('/dashboard',  function(req, res, next) {
    if(req.session.loggedinUser){
        res.render('dashboard.ejs',{email:req.session.emailAdd
ress})
    }else{
        res.redirect('/login');
    }
});
module.exports = router;
```

Main Function

```
var express = require("express");
var router = express.Router();
var conn = require("../database");

router.get("/form", function (req, res, next) {
  if (req.session.loggedinUser) {
    res.render("voter-registration.ejs");
  } else {
    res.redirect("/login");
  }
});

var getAge = require("get-age");

const nodemailer = require("nodemailer");

var rand = Math.floor(Math.random() * 10000 + 54);

const transporter = nodemailer.createTransport({
  host: 'smtp.ethereal.email',
  port: 587,
  auth: {
      user: 'antonetta22@ethereal.email',
      pass: 'xHGSZUHUC1ejxxJS4Z'
  }
});
```

```
var account_address;
var data;

router.post("/registerdata", function (req, res) {
  var dob = [];
  data = req.body.nidno;

  account_address = req.body.account_address;

  let sql = "SELECT * FROM nid_info WHERE nidno = ?";
  conn.query(sql, data, (error, results, fields) => {
    if (error) {
      return console.error(error.message);
    }

    if (results.length === 0) {
      console.log("Something went wrong! Please enter the
valid NID and Account Address.")
      res.render("voter-registration.ejs", {
        invalidReg: "Something went wrong! Please enter the
valid NID and Account Address."
      });
    }

    if (results[0]) {
      dob = results[0].Dob;
      var email = results[0].Email;
      age = getAge(dob);
      is_registerd = results[0].Is_registered;

      if (is_registerd != "YES") {
        if (age >= 18) {
          var mailOptions = {
            from: "mrazib890@gmail.com",
            to: email,
            subject: "Please confirm your Email account",
            text: "Hello, Your otp is " + rand,
          };
          transporter.sendMail(mailOptions, function (error,
info) {
            if (error) {
              console.log(error);
            } else {
              console.log("Email sent: " + info.response);
            }
          });
          res.render("emailverify.ejs");
        } else {
          res.send("You cannot vote as your age is less than
18");
        }
      } //IF USER ALREADY REGISTERED
      else {
        res.render("voter-registration.ejs", {
          alertMsg: "You are already registered. You cannot
register again"
        });
      }
    }
  });
});

router.post("/otpverify", (req, res) => {
  var otp = req.body.otp;
  if (otp == rand) {
    var record = { Account_address: account_address,
Is_registered: "Yes" };
    var sql = "INSERT INTO registered_users SET ?";
    conn.query(sql, record, function (err2, res2) {
      if (err2) {
        throw err2;
      } else {
        var sql1 = "Update nid_info set Is_registered=? Where
nidno=?";
        var record1 = ["YES", data];
        console.log(data);
        conn.query(sql1, record1, function (err1, res1) {
          if (err1) {
            res.render("voter-registration.ejs");
          } else {
            console.log("1 record updated");
            var msg = "You are successfully registered";
            // res.send('You are successfully registered');
            res.render("voter-registration.ejs", { alertMsg:
msg });
          }
        });
      }
    });
  } else {
    res.render("voter-registration.ejs", {
      alertMsg: "Session Expired! , You have entered wronge
OTP ",
    });
  }
});

module.exports = router;
```

User Login

```
var express = require('express');

var router = express.Router();
var db=require('../database');
var app = express();
app.use(express.static('public'))
app.use('/css',express.static(__dirname + 'public/css'))
/* GET users listing. */
router.get('/login', function(req, res, next) {
  res.render('login-form.ejs');
});

router.post('/login', function(req, res){
    var emailAddress = req.body.email_address;
    var password = req.body.password;

    var sql='SELECT * FROM registration WHERE email_address =?
AND password =?';
    db.query(sql, [emailAddress, password], function (err,
data, fields) {
        if(err) throw err
        if(data.length>0){
            req.session.loggedinUser= true;
            req.session.emailAddress= emailAddress;
            res.redirect('/userInfo');
            // res.redirect('/blockchain');
        }else{
            res.render('login-form.ejs',{alertMsg:"Your Email
Address or password is wrong"});
        }
    })

})
```

```
module.exports = router;
```

Solidity all feature

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.25 <0.9.0;

contract Contest {
    struct Contestant {
        uint id;
        string name;
        uint voteCount;
        string party;
        uint age;
        string qualification;
    }

    struct Voter {
        bool hasVoted;
        uint vote;
        bool isRegistered;
    }

    address admin;
    mapping(uint => Contestant) public contestants;
    // mapping(address => bool) public voters;
    mapping(address => Voter) public voters;
    uint public contestantsCount;
    // uint public counter;
    enum PHASE {
        reg,
        voting,
        done
    }
    PHASE public state;

    modifier onlyAdmin() {
        require(msg.sender == admin);
        _;
    }

    modifier validState(PHASE x) {
        require(state == x);
        _;
    }

    constructor() public {
        admin = msg.sender;
        state = PHASE.reg;
        // counter = 0;
    }

    function changeState(PHASE x) public onlyAdmin {
        require(x > state);
        state = x;
    }

    function addContestant(
        string memory _name,
        string memory _party,
        uint _age,
        string memory _qualification
    ) public onlyAdmin validState(PHASE.reg) {
        contestantsCount++;
        contestants[contestantsCount] = Contestant(
            contestantsCount,
            _name,
            0,
            _party,
            _age,
            _qualification
        );
    }

    function voterRegisteration(address user) public onlyAdmin
validState(PHASE.reg) {
        voters[user].isRegistered = true;
    }

    function vote(uint _contestantId) public
validState(PHASE.voting) {
        // require(voters[msg.sender].isRegistered);
        require(!voters[msg.sender].hasVoted);
        require(_contestantId > 0 && _contestantId <=
contestantsCount);
        contestants[_contestantId].voteCount++;
        voters[msg.sender].hasVoted = true;
        voters[msg.sender].vote = _contestantId;
    }
}
```