A

Project Report

on

ONLINE VOTING SYSTEM USING BLOCKCHAIN

Submitted in partial fulfillment for the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION TECHNOLOGY

By

E. R. Rishwanth Reddy	**(245118737049)**
C. Sumanth	**(245118737074)**
K. Nivas Teja	**(245118737080)**

Under the guidance of

D. B. V. Ravisankar

Associate Professor

Department of IT

MATURI VENKATA SUBBA RAO ENGINEERING COLLEGE

(An Autonomous Institution)

Department of Information Technology,

(Affiliated to Osmania University, Hyderabad)

Nadergul, Hyderabad, TELANGANA – 501510

Academic year: 2021-22

MATURI VENKATA SUBBA RAO(MVSR)
ENGINEERING COLLEGE
(An Autonomous Institution)
(Affiliated to Osmania University, Hyderabad, Recognized by AICTE)
Nadergul, Saroornagar Mandal, Hyderabad-501510

DEPARTMENT OF INFORMATION TECHNOLOGY
CERTIFICATE

*This is to certify that the project work entitled "**Online Voting System using Blockchain**" is a bonafide work carried out by **Mr. Eleti Raju Rishwanth Reddy (2451-18-737-049), Mr. Chilkuri Sumanth (2451-18-737-074), Mr. Korra Nivas Teja (2451-18-737-080)** in fulfillment of the requirements for the award of degree of **Bachelor of Engineering** in **Information Technology** from **Maturi Venkata Subba Rao Engineering College,** affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2021-22. under our guidance and supervision.*

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of Project Coordinator

Signature of Guide

Signature of Head, ITD

Signature of External Examiner

DECLARATION

This is to certify that the work reported in the present project entitled "Online Voting System using Blockchain" is a record of bona fide work done by us in the Department of Information Technology, Maturi Venkata Subba Rao Engineering College, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Roll Number:	Student Name:	Signature of the Student
(2451-18-737-049)	E. R. Rishwanth Reddy	
(2451-18-737-074)	C. Sumanth	
(2451-18-737-080)	K. Nivas Teja	

ACKNOWLEDGMENT

We would like to express our sincere gratitude and indebtedness to our project guide **Mr. D.B.V.Ravisankar, Associate Professor** for his valuable suggestions and interest throughout the course of this project.

We would like to extend our gratitude to **Dr. D. Shanthi**, **Associate Professor,** Department of Information Technology, Maturi Venkata Subba Rao Engineering College, for her valuable suggestions and timely help during the course of the project.

We are also thankful to **Dr K VenuGopal Rao,** Dean – Academics & Head, Department of Information Technology, Maturi Venkata Subba Rao Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (IT).

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

E. R. Rishwanth Reddy (2451-18-737-049)

C. Sumanth (2451-18-737-074)

K. Nivas Teja (2451-18-737-080)

Maturi Venkata Subba Rao Engineering College

Department of Information Technology

COURSE NAME: Project Work II

COURSE CODE: PW861IT

VISION

To impart technical education to produce competent and socially responsible engineers in the field of Information Technology.

MISSION

a. To make teaching learning process effective and stimulating.
b. To provide adequate fundamental knowledge of sciences and Information Technology with positive attitude.
c. To create an environment that enhances skills and technologies required for industry.
d. To encourage creativity and innovation for solving real world problems.
e. To cultivate professional ethics in students and inculcate a sense of responsibility towards society.

PROGRAM EDUCATIONAL OBJECTIVES(PEOS)

The Program Educational Objectives of undergraduate program in Information Technology are to prepare graduates who will:

I. Apply knowledge of mathematics and Information Technology to analyze, design and implement solutions for real world problems in core or in multidisciplinary areas.

II. Communicate effectively, work in a team, practice professional ethics and apply knowledge of computing technologies for societal development.

III. Engage in Professional development or postgraduate education to be a life-long learner.

(A) PROGRAM OUTCOMES(POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principle and apply 6 these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Lifelong learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

(B) PROGRAM SPECIFIC OUTCOMES (PSOs)

1. **Hardware design:** An ability to analyse, design, simulate and implement computer hardware / software and use basic analogue/digital circuits, VLSI design for various computing and communication system applications.
2. **Software design**: An ability to analyse a problem, design algorithm, identify and define the computing requirements appropriate to its solution and implement the same.

COURSE OBJECTIVES AND OUTCOMES

Course Objectives

1. To enhance practical & Professional skills.
2. To familiarize the tools and techniques of symmetric literature survey and documentation.
3. To expose students to industry practices and teamwork.
4. To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes

On successful completion of this course student will be

1. Define a problem of the recent advancements with applications towards society.
2. Outline requirements and perform requirement analysis for solving the problem.
3. Design and develop a software and/or hardware, based solutions within the scope of project using contemporary technologies and tools.
4. Test and deploy the applications for use.
5. Develop the Project as a team and demonstrate the application, with effective written and oral communications

ABSTRACT

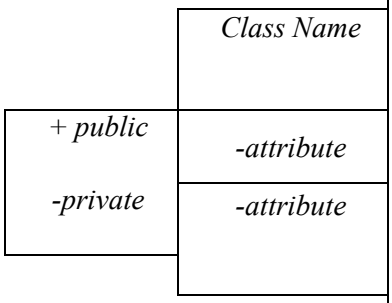
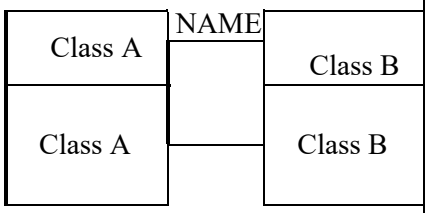
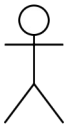
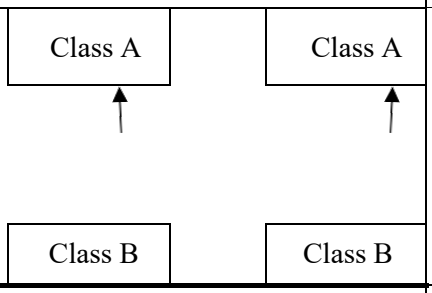
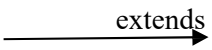
An online Voting System is an application for conducting elections using any internet- enabled device and without a traditional paper ballot system. Various concerns arise regarding the reliability and security of this system. Blockchain is the solution for most of the concerns. Using blockchain, we secure transactions of the online voting system. Making the system decentralized and immune to modifications. This system will overcome most of the disadvantages of the Balloting System and traditional Electronic System which include lack of transparency, fake votes, political manipulation, physical presence to vote, etc.

We aim to develop two client-side applications for android and the web through which an authorized user can cast his/her vote from anywhere in the world. Which will be stored as blocks and form a blockchain making the system decentralized and secure. Through this system, we can reduce costs, improve convenience, security, and reliability.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Association represents static relationships between classes. Role represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single class.
4.	Aggregation		Interaction between the system and external environment
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.

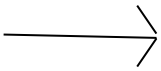
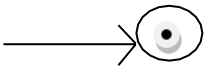
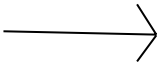
8.	State	State	State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states
12.	Metamask		Metamask is a browser extension that works as wallet for Ethereum based transaction.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Elections are the major contributing factor for democracy, they play a crucial role in a country's future. Elections have always been a controversial topic in India, starting from the balloting system to the electronic voting system, there have always been loopholes in the system like counting errors, fake votes, lack of transparency, vulnerability to hacking, political manipulations, a requirement of the physical document and physical presence of the voter, and others. These problems can be solved by making the entire voting system online. Using an online voting system, a voter can cast their vote from anywhere on the planet by just using any internet-enabled device, this makes elections easier to conduct and has the potential to increase voting percentage. In this modern-day, almost everyone has access to an internet-enabled device, using any of the identification documents, an eligible voter can authenticate them self and cast their vote. Storage and counting of votes are done automatically so the results can be announced as soon as possible.

Electronic and online voting system can provide more security and integrity than EVM. It also provides privacy to the users as the eligible voters can cast their votes using their computers or even smartphones, thus maintaining anonymity. It also builds trust on the system as it is online and completely an open system. It can also increase the number of participants' involvement. Blockchain eliminates the need of a central server to manage network and a centralized database thus ensuring trust. It is a complete decentralized open ledger system. The public ledger records all the votes casted and is permanent and immutable. It ensures that no vote can be changed once casted. If someone tries to manipulate the ledger, they need to first hack all previous blocks before adding new block, which is nearly impossible because of consensus mechanism. To comprise the network hacker should compromise at least one-third or sometimes half of the network depending on the consensus employed.

1.1 PROBLEM STATEMENT

Elections are a requirement by any organization to select a person among a group of people. Not every organization can afford additional hardware and efforts to conduct simple internal elections. Using an online voting system, eligible voters of an organization can access the election portal through an internet-enabled device login and authenticate themself and cast their vote. Storage and counting of votes are done internally so, after elections, anyone can view the results, making the entire election system transparent. Through this, organizing and conducting elections becomes easy and convenient.

1.2 OBJECTIVES

It eliminates the need to print ballot papers or open polling stations-voters can vote from wherever there is an Internet connection. Despite these benefits, online voting solutions are viewed with a great deal of caution because they introduce new threats.

1.3. MOTIVATION FOR BLOCKCHAIN TECHNOLOGY

Blockchain is a complex technology for secure transactions, also described as a distributed decentralized ledger. Blockchain was invented by Satoshi Nakamoto, in 2008 for Bitcoin cryptocurrency. By design, Blockchain is resistant to modification of its data. Blockchain is implemented in various sectors like Banking and financial services, Internet, Healthcare, Automotive, Media and entertainment, Travel, and transportation, etc.

Traditionally, the database is maintained by a central authority or a single organization that has then complete control of the database. It has the ability to tamper with the database and manipulate the data. Usually, the authority maintaining the database is the same that has created it and will be using it. In such cases the organization has no motive of manipulating or falsifying its own data. But in other cases, involving financial matters or sensitive data like voting, it's not wise to give total control of database to a single authority or organization. Even if the organization is guaranteed as to not make any fraudulent changes to the database, it is easier for hackers to manipulate a central database. To avoid such situations, blockchain makes the database public, so that anyone can store an individual copy of database that can always be compared to check for manipulations.

However, the individual copies must always be updated to maintain consistency. To maintain a consistent decentralized database, blockchain utilizes consensus mechanism.

1.4. EXISTING SYSTEM

Centralized Electronic Voting System

There has been a new record set by India for voter turnout at 66.38% for the General Election 2014 and awareness to vote is rising. Since then, the need of project to create an electronic voting system which is centralized i.e., the voter can vote from any region and data will be automatically uploaded on a centralized server has become a necessity. Number of voters increases day-by-day as time and population in developing countries increase in decades. In day-to-day practice, majority of voters are busy or occupied to go for work and most of the voter's homes are situated far away from the voting centers, also voters don't like to wait in queues as their time is also valuable. Because of such reasons voter don't visit the polling booth and percentage of voting is decreasing. Centralized

server-based voting system is very similar for the amendment of this percentage of voting, which is every person's right. In ancient voting systems such as the electronic voting and paper-based voting, there were issues of security and the time taken to count the votes were more. Some improvements are needed in this field. The idea of Centralized Electronic Voting System can prove very useful to solve these problems. i.e., Rigging and Security problems are reduced as compared to old system. Because of the Centralized Electronic Voting System, the problem of Rigging i.e., Fake Voting and security issues are solved. The voter can cast his vote from any region because of this there is no need to go to particular region for voting and due to this time is saved. As well as the simultaneous counting can be done in this system and because of this time required to display results is less. In this system a database for registration of voter (i.e., Add, Search and Edit the voter's information in this database) is created. The authentication is done by fingerprint scanner. Authentication of the voter is done by matching his data from the previously made centralized data server. If the fingerprint gets matched, then that person can cast his vote and therefore this data is saved and uploaded on the server at the end of Election Day. As the data is centralized i.e., the data would also be uploaded from different location for the same region. Simultaneous counting is also done, and the result would be displayed on LCD after the polling officer enters the password. There is great need for implementation of this system on large scale as it reduces manufacturing cost, voter's efforts and vote counting time. Centralized Electronic Voting System has been designed for the sole purpose of minimizing hardware and creating a better interface with the user by making this system more reliable.

1.5. PROPOSED SYSTEM

The proposed security system has been developed with an intention to prevent tampering, manipulation. Online voting system use decentralized blockchain network, no one has to know or trust anyone else. Each member in the network has a copy of the exact same data in the form of a distributed ledger. If a member's ledger is altered or corrupted in any way, it will be rejected by the majority of the members in the network. Using blockchain, voting process can be made more secure, transparent, immutable, and reliable. We developed a web-page where election is conducted, admin and eligible voter has the access to open the web-page by account address and hash value of ganache account. Admin got initialized to start election by filling out details, add candidates who want to participate in election and approve the voters who are eligible. Voter has to register with ganache account address, name and phone number and cast votes. Results are displayed after ending the election only by admin.

In this project we use solidity for admin and voter functions and for transaction of Ethereum coin.

1.6. SCOPE OF THE PROJECT

- Any organization that wants to conduct elections can use this as an effortless and secure solution.
- This solution can be further improved for government election purposes.

1.7. OUTLINE OF THE RESULTS

1.7.1. Election heads can assign default credentials, create and organize elections.

1.7.2. A voter can cast his vote transparently and from anywhere using an internet-enabled device.

1.7.3. No additional counting or paperwork is required to announce the winner.

1.7.4. Everyone in the organization can view the election results.

CHAPTER 2

LITERATURE SURVEY

Sl No.	Name	Author's Name & Year	Technology Used	Advantages	Disadvantage
1.	Development of Microcontroller Based Electronic Voting Machine(EVM)	Meharaj Unnisa Mr.Soumen Ghosh (January 2014)	Microcontroller Based Electronic Voting Machine.	Data storage capacity is More.	Complexity in Hardware Implementation .
2.	Smart wireless authenticating voting machine	Devendra Vijay Naik Firas I. Hazzaa (2015)	Fingerprint Technology and Web Based.	Security and Accuracy is More.	Complexity in Implementation.
3.	Design of GSM Based Electronic Voting Machine with Voter Tracking	Vaibhav Bhatia, Rahul Gupta (June 2015)	GSM Technology	There is no need to go to the Polling Booth for voting.	Range problem can occur in GSM Technology
4.	AADHAR based Electronic Voting Machine	R.Murli Prasad M.Venkata Rao Ankita R. Kasliwal (June 2016)	Aadhar Card Based Electronic Voting System	There is no need to create new database.	Aadhar Card data is confidential.
5.	Raspberry Pi voting system, a reliable technology for transparency in democracy	Dimesh Bommisetty Md.Mami nul Islam G.Keerthana (December 2016)	RFID, Image Processing Technology and IOT Based	Because of IOT Technology data is securely stored in the Cloud.	RFID/ Smart Card can be misplaced and because of Raspberry Pi cost is more.

6.	Electronic Voting Machine using Bio-metric Finger Print with Aadhar Card Authentication	Shekhar Mishra, Firas Hazzaa (March 2017)	Biometric Technology	Because of Fingerprint identification Security increases and Rigging stopped	Person has to present In the voting booth for voting. Because of this time and money is waste.

2.2. PROPOSED SYSTEM

ETHEREUM BASED SOLUTION

Blockchains can be classified into two categories: permission-ed and permission-less. Permission-ed blockchains are private blockchain network with restrictions on participation. Permission-less blockchains are public in nature. In public blockchains there are no restrictions such that anyone can read or write on blockchain ledger database. Ethereum is a public decentralized blockchain network.

Basically, Ethereum is platform that allows programmers to build decentralized applications using blockchain technology. It is a permission-less blockchain network. This section describes the types of accounts used in Ethereum. Ethereum has two account types:

1. External Accounts,
2. Contract Accounts

An externally owned account is a user-controlled account. It represents an external agent of network like users, miners etc. These accounts are regulated with a public-private key cryptography like RSA algorithms. Mainly external accounts are used by users as a means to interact with the Ethereum blockchain.

A contract account is a smart contract which is a collection of code that regulates blockchain. These are stored at a specific address, hence considered as accounts. Contract accounts are always either invoked by some external accounts or by other contract accounts. These contracts are written in high level scripting languages such as Solidity and Serpent.

Both of these accounts can store Ether. Ether is the crypto currency of Ethereum, denoted by “ETH” in crypto currency exchanges. It is used for transactions fee and services in the Ethereum network. These are used to pay Gas or transactions done. Gas is an intermediary token used to make

payment for computational work done for executing a smart contract or for some transactions. Gas can be purchased using Ether.

The idea behind an e-voting system based on blockchain is similar to digital wallets. The system or authority can issue a digital wallet to each participant after verifying its identity. The wallet issued must contain the user credentials and a single coin representing a single chance to vote. When a user selects a candidate and casts its vote, the coin in the user's wallet is transferred to candidate's account or wallet. At the end the number of coins in each candidate's wallet represents the number of votes cast to him.

This section discusses the design considerations for implementing an electronic voting system based on blockchain. Later, an overview of Ethereum and smart contract technology is presented.

Design considerations-

Following points are important and should be considered while implementing e-voting system:

- The e-voting system should verify the identity of voters and authenticate only eligible voters.
- The e-voting system should not allow access to invalid candidates.
- Any voter should get only a single chance to vote i.e. system should prevent double voting.
- It should provide complete privacy to voters and the votes should not be traceable.
- It should not allow tampering with the votes casted by anyone.
- The system should not allow single authority control on counting.

2.2. ALGORITHMS

2.2.1. SHA-256

(Secure Hash Algorithm 256) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA) and first published in 2001. They are built using the Merkle–Damgård construction, from a one-way compression function itself built using the Davies–Meyer structure from a specialized block cipher. We use SHA256 to hash user's password and store it in our database. SHA256 generates output of 256 bits. This hash is used during log in to authenticate a user.

2.3.2 SMART CONTRACT

A smart contract is a computer program or transaction protocol that is designed to automate execution, control, and documentation of legally significant events and activities in accordance with the conditions of a contract or agreement. Below are the smart contract code snippets for main application modules.

CHAPTER 3
SYSTEM REQUIREMENTS SPECIFICATIONS

3.1. HARDWARE REQUIREMENT

- Processor: Minimum 1.7 GHz; recommended 2GHz (or more)
- Ethernet Connection (LAN of Wi-Fi)
- Hard Drive: 100GB or more
- Memory (RAM): 4GB or more

3.2. SOFTWARE REQUIREMENTS/ TOOLS AND TECHNOLOGIES

3.2.1. VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a free source code editor made by Microsoft for Windows, Linux and macOS. It provides built-in support for JavaScript, TypeScript and Node.js. You can add extensions to provide support for numerous other languages such as C++, C#, Java, Python, PHP, and Go, and runtimes such as .NET and Unity.

3.2.2. NODE.JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the Chrome 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.

3.2.3. SMART CONTRACT

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

3.2.4. SOLIDITY

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

3.2.5. META MASK

Meta-Mask 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.

3.2.6. 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 dApps in a safe and deterministic environment. Ganache comes in two flavors: a UI and CLI.

3.2.7. 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.

CHAPTER 4

SYSTEM DESIGN

4.1. ARCHITECTURE

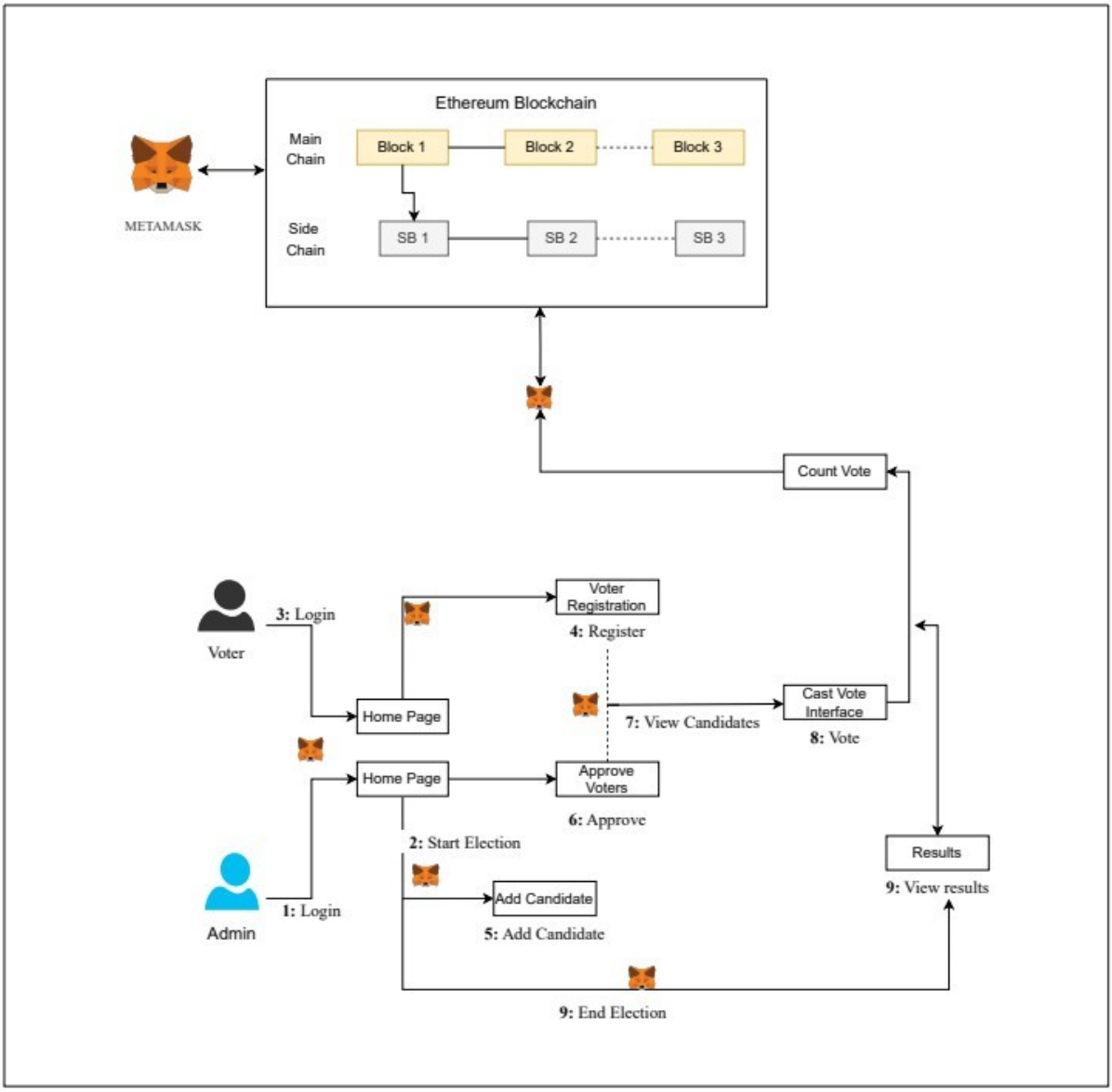


Figure 4.1 : Architecture

- In the first step the admin will login to webpage by account address of ganache account.
- Admin will create election by filling out details and start the election.
- Next, adding of candidates who want to participate in election.
- Voter should register with ganache account address, name, and phone number.
- Admin will verify voter who registered and approve if voter is eligible.

- Now eligible voter can caste vote only once.
- When the admin ends the election, then only results are displayed.
- All the transaction data stored in blockchain.

4.2. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

4.2.1. USE CASE DIAGRAM

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. The online voting system has two actors, Voter and Admin. Common modules for these actors are login and view results both Voter and Admin can log in and check results of ongoing elections. Additionally, a Voter has to register and has a cast vote module to cast his/her vote in an eligible election. Admin can create elections, add candidates and approve voters.

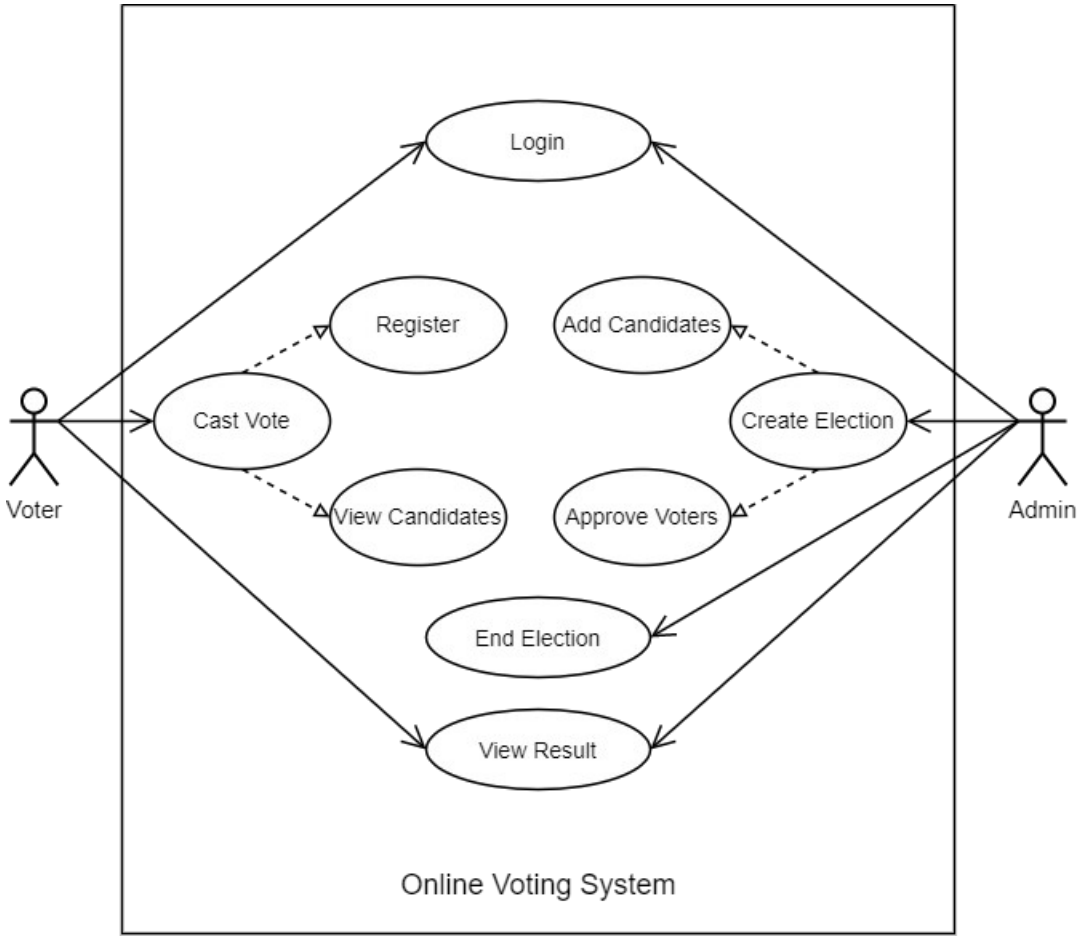


Figure 4.2: Use Case Diagram

4.2.2. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains which information.

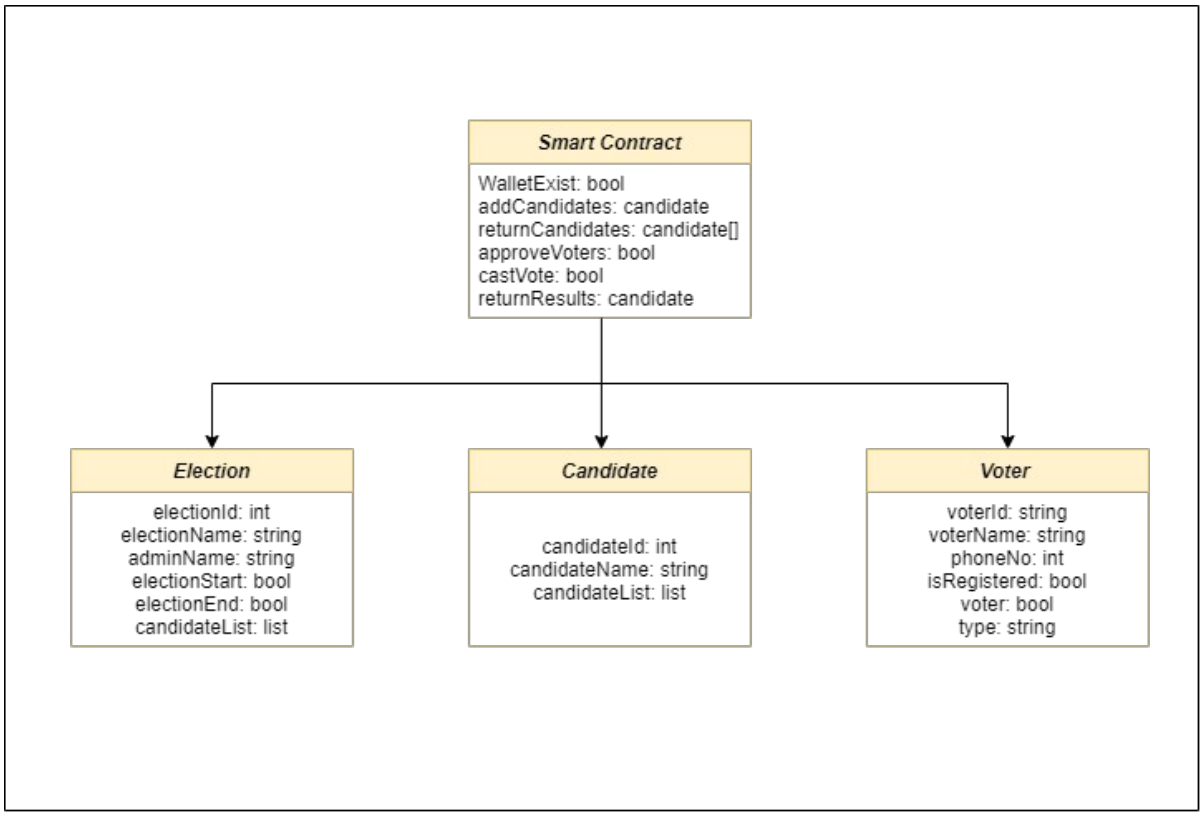


Figure 4.3: Class Diagram

4.2.3. SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

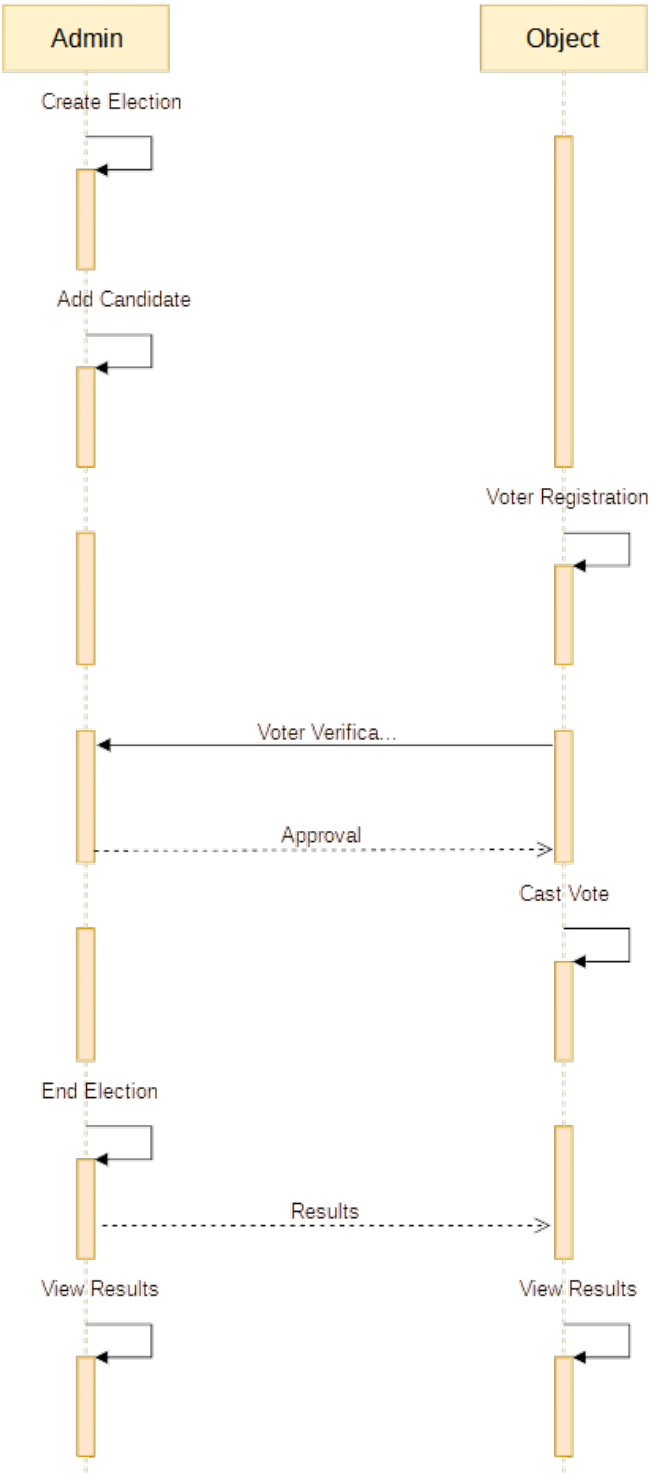


Figure 4.4: Sequence Diagram

4.2.4. COLLABORATION DIAGRAM

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

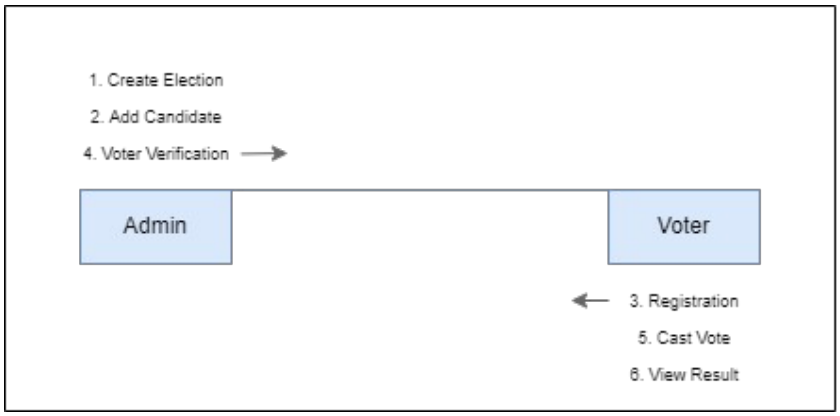


Figure 4.5: Collaboration Diagram

4.2.5. DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

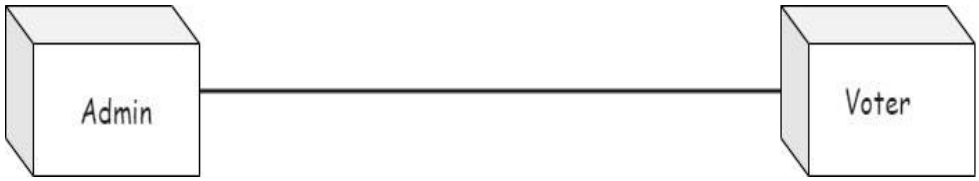


Figure 4.6: Deployment Diagram

4.2.6. ACTIVITY DIAGRAM

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

For our application, online voting system. We have two activity diagrams one for Voter and the other for Admin. A user identifies as a voter or admin using their ganache account (hash value).

An Admin has access to five modules to create and manage elections. Create Election, is used to create an election. Admin has to input election type, organization name. After Creating an election, candidates competing in the election are added. This is done using Add Candidate module. Admin will choose an election and input candidate details like candidate name and description. After submitting, the candidates are added to the election. Multiple candidates can be added to an election. Approve Voter module is used to approve a voter who is eligible to vote for a specific election. Initially Admin sends private key of ganache account which connects them to meta mask. Then, Admin needs input details of a voter which include account address which match the private key of that account and name, phone number and the election for which the voter is eligible to cast vote. Admin also has View Result and Log out modules similar to the voter.

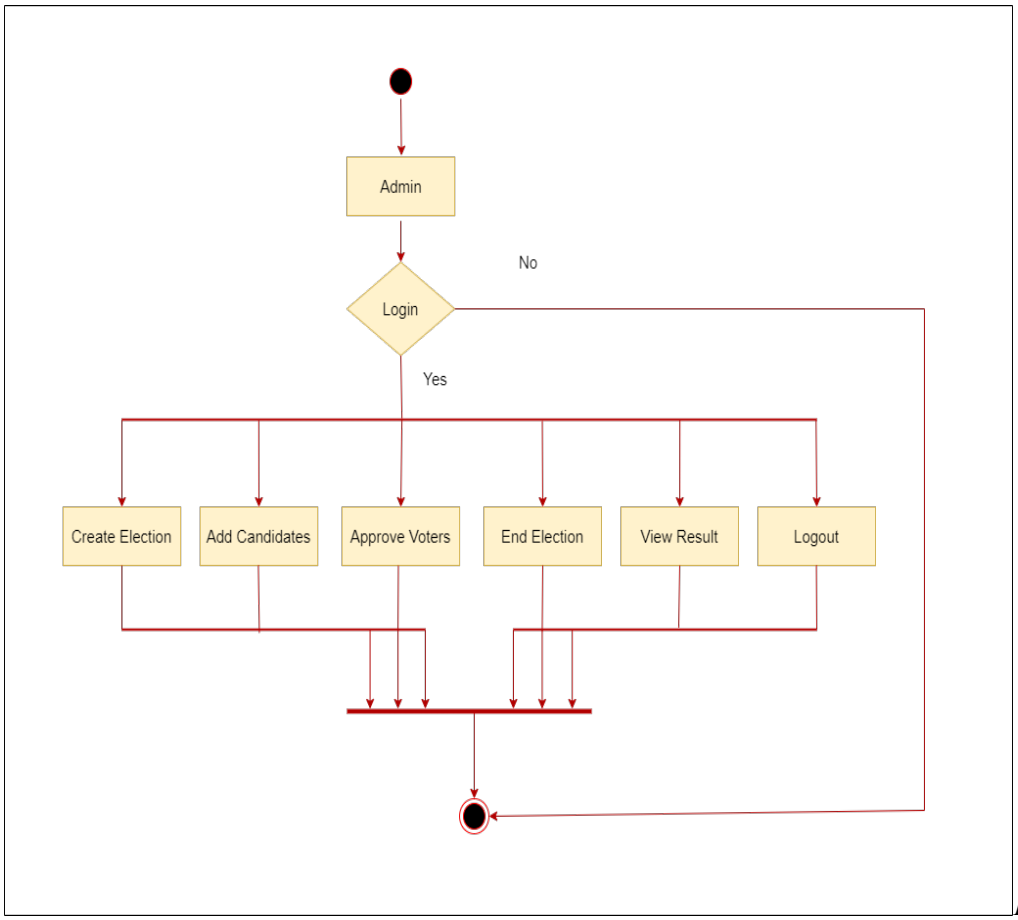


Figure 4.7: Admin Activity Diagram

A Voter has access to four modules, Register, Cast Vote, View Result, and Logout. In Register, voter needs to enter account address of his ganache account, name and phone number. In Cast Vote, Data of the voter, and candidates competing in the election are fetched. The voter can choose and submit his / her vote. Internally conditions like previously voted are checked before a vote is submitted. Voters can view the results of the elections conducted so far and log out after using the application.

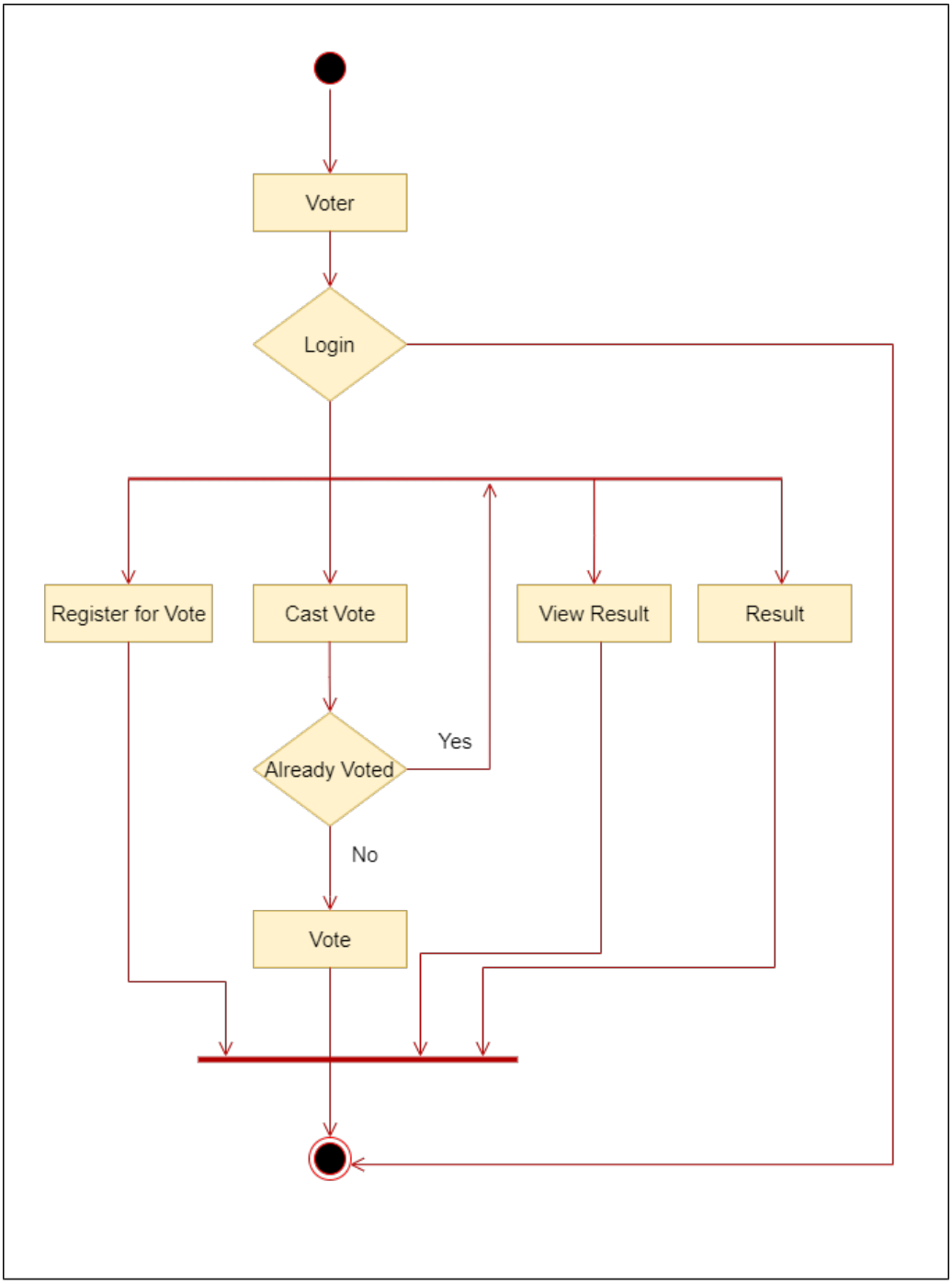


Figure 4.8: Voter Activity Diagram

4.2.7 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

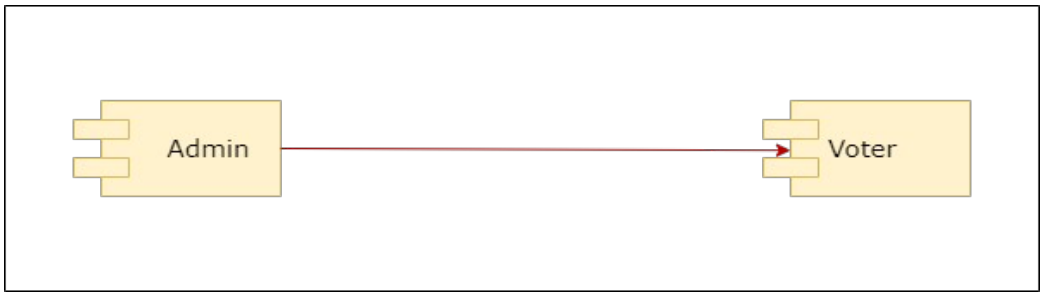


Figure 4.9: Component Diagram

CHAPTER 5

IMPLEMENTATION

5.1 ENVIRONMENTAL SETUP

Installing Visual Studio Code:

1. To download and install VS Code visit the official website of VS Code https://code.visualstudio.com/

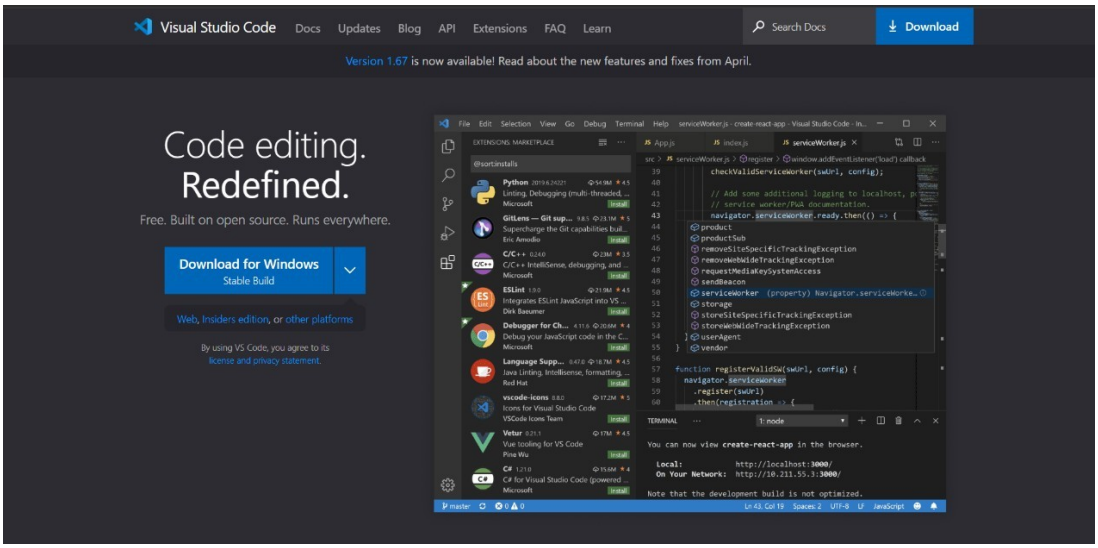


Figure 5.1: VS Code

2. Once the download is complete, run the exe for installing VS Code. Now click on install now.
3. You can see VS Code installing at this point.
4. When it finishes, you can see a screen that says the setup was successful.

Installing Node.js:

1. To download Node.js visit official website https://nodejs.org/en/ and click on download (LTS recommended).

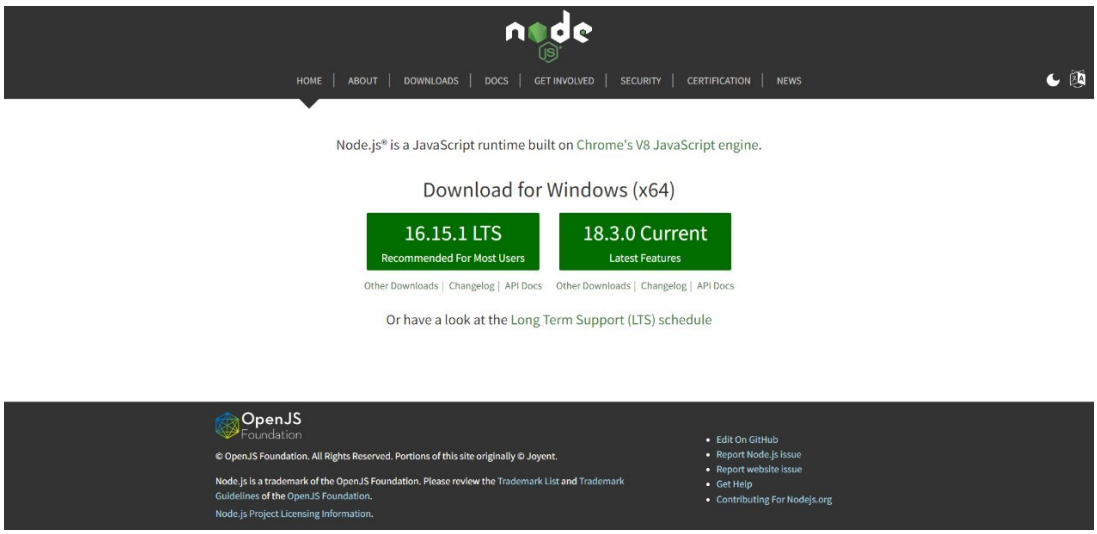
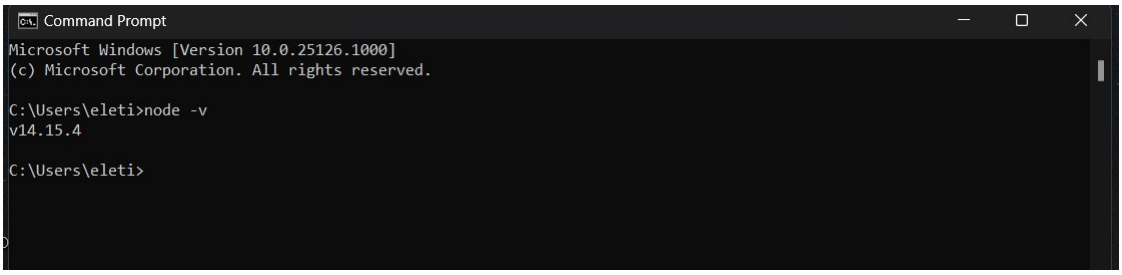


Figure 5.2: node.js

2. Once the download complete, run the exe for installing Node.js.
3. Select “Add to path” and click “Next”.
4. You can see node.js installing at this point.
5. When it finishes, you can see a screen that says the setup was successful. Now click on “Close”.
6. To confirm installation was successful, open Command Prompt and type “node -v”



Installing Ganache:

1. To install Ganache visit official website https://trufflesuite.com/ganache/ and click on Download Button.

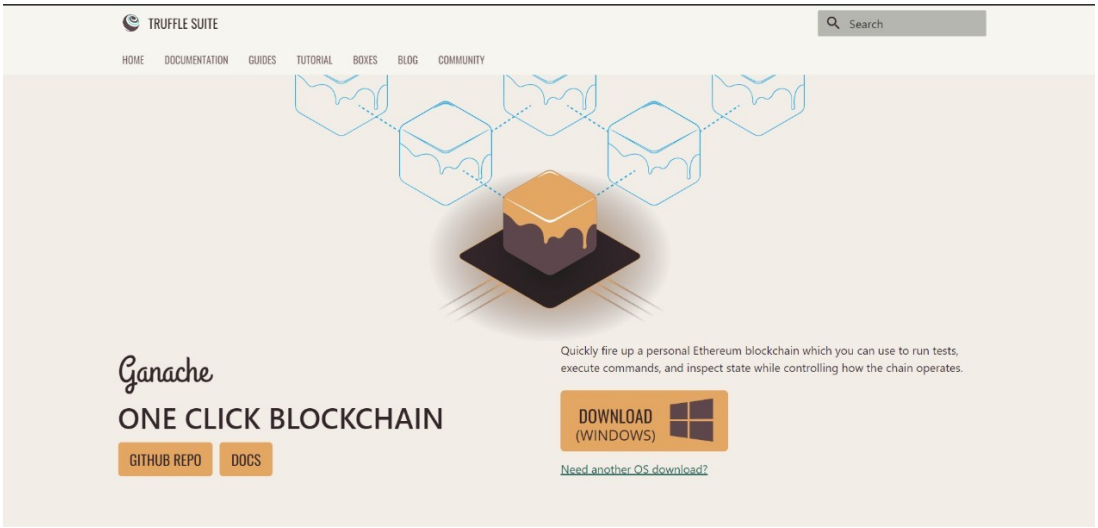


Figure 5.3: Ganache

2. Once the download complete, run the exe for installing Ganache.
3. You can see node.js installing at this point.
4. When it finishes, you can see a screen that says the setup was successful. Now click on “Close”.

Using Remix:

1. Remix IDE is an open source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix is used for the entire journey of contract development with Solidity language as well as a playground for learning and teaching Ethereum.
2. Remix IDE has modules for testing, debugging and deploying of smart contracts and much more.

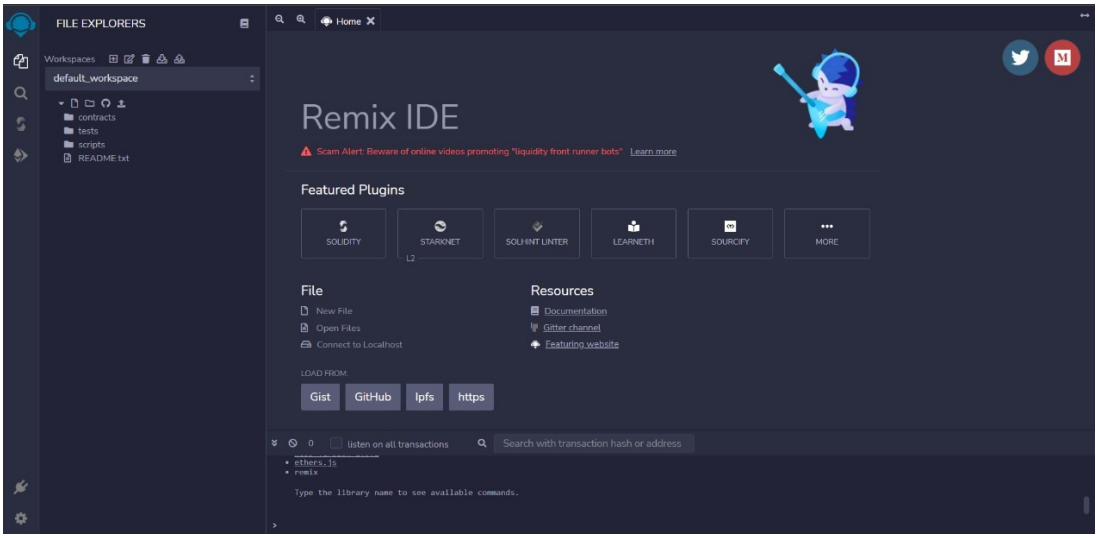


Figure 5.4: Remix IDE

Metamask:

1. Metamask is a browser add-on(browser extension) that manages a user's Ethereum wallet by storing their private key on their browser's data store and the seed phrase encrypted with their password.
2. It is a non-custodial wallet, meaning, the user has full access and responsibility their private key. Once lost, the user can no longer control the savings or restore access to the wallet.

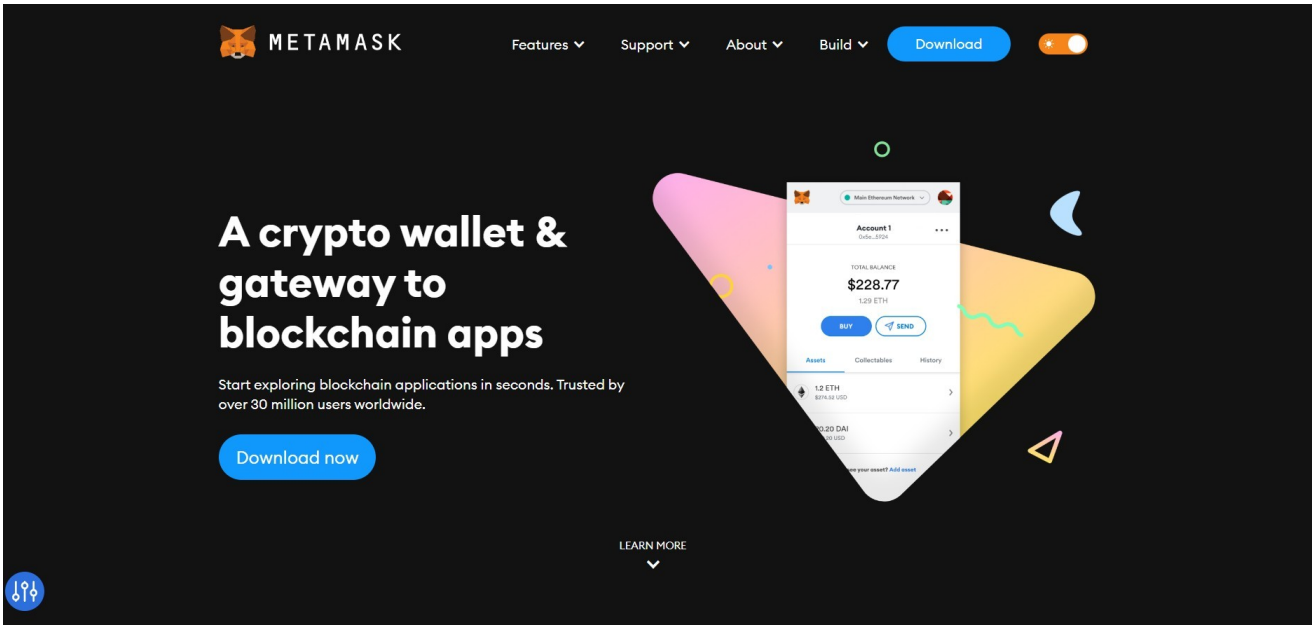


Figure 5.5: Metamask

5.1 MODULE DESCRIPTION

5.1.1 LOG IN

The user is required to log into the application to cast his / her vote or create elections. Both admin and voter have a common login page were based on their role, different dashboards are displayed. To log in, a user needs to enter his / her email and password. If the provided credentials are correct, the user will be logged in, and further operations can be performed.

5.1.2 VIEW RESULT

View Result is another common module for Voter and Admin. Results of the elections are shown in this module. A user can select currently running or previously completed elections to view its result. The result is shown on a bar graph with votes received for each candidate.

5.1.3 CAST VOTE

The Cast Vote module is accessible only to a voter. This module will allow a voter to cast his / her vote in an ongoing election. All the parties competing in the election and their descriptions are available while voting. The voter can choose his / her choice and submit. Once a vote is submitted, changes cannot be made, and a voter cannot vote multiple times.

5.1.4 CREATE ELECTION

Create Election module is only accessible to admin. This module is used to create elections. Multiple elections can be running simultaneously. Admin needs to provide details like election name, start date, and end date to create an election. The election will start and end automatically by admin provided start and end date.

5.1.5 CREATE PARTY

Create Party module is used to add parties competing for an election. This module is only accessible to the admin, and details like party name and description are required to create and add a party to an election. All the parties added to an election are visible to the voter and can make his / her choice while casting their vote.

5.1.6 CREATE VOTER

Create Voter module is used to create and add eligible voters to an election. This module is only accessible to the admin. To create a voter, the admin needs to provide details like first name, last name, email, and election he/she is eligible to vote. A random password is generated and mailed to the given email of the voter. This password will be used during login by the voter. Admin creating a voter has no access to the voter's password.

5.1.7 LOG OUT

Log out is a common module for voters and admin. This module is used to log out from the application and end the current session.

CHAPTER 6
RESULTS (SCREENSHOTS)

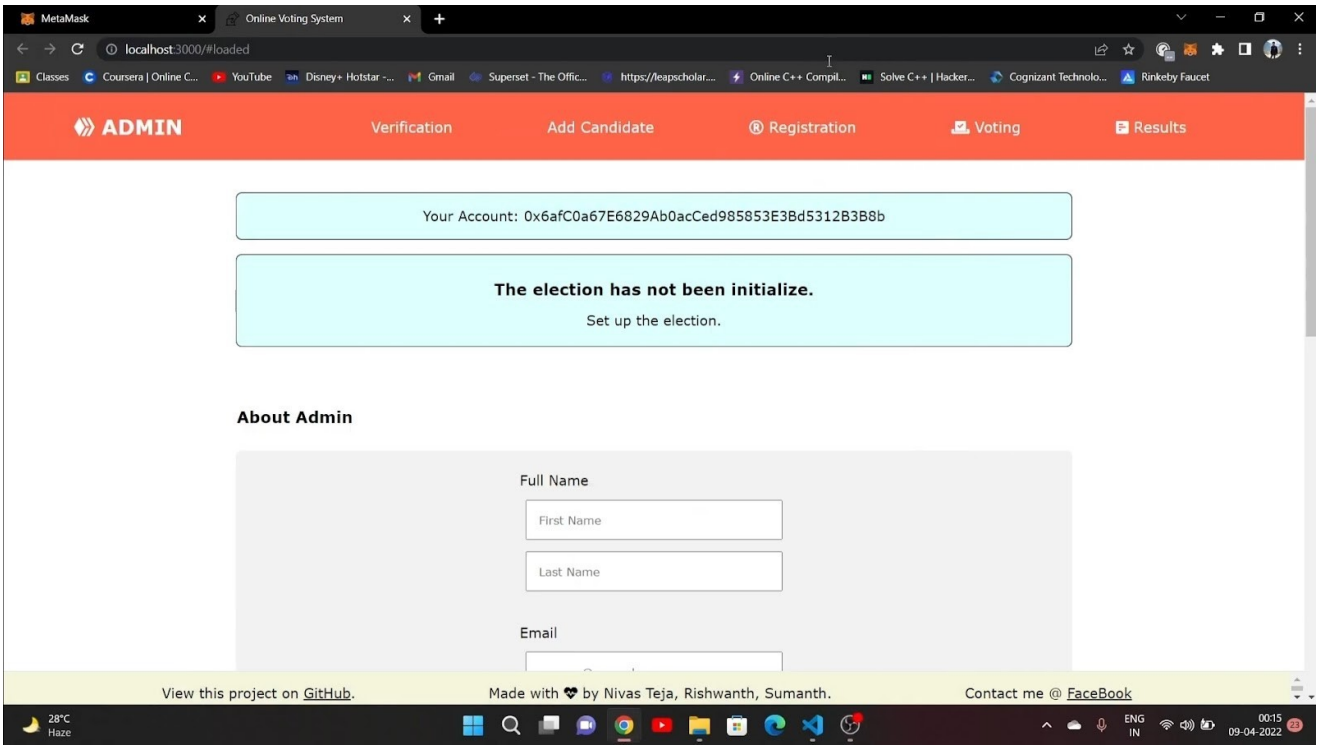


Figure 6.1: Admin Page

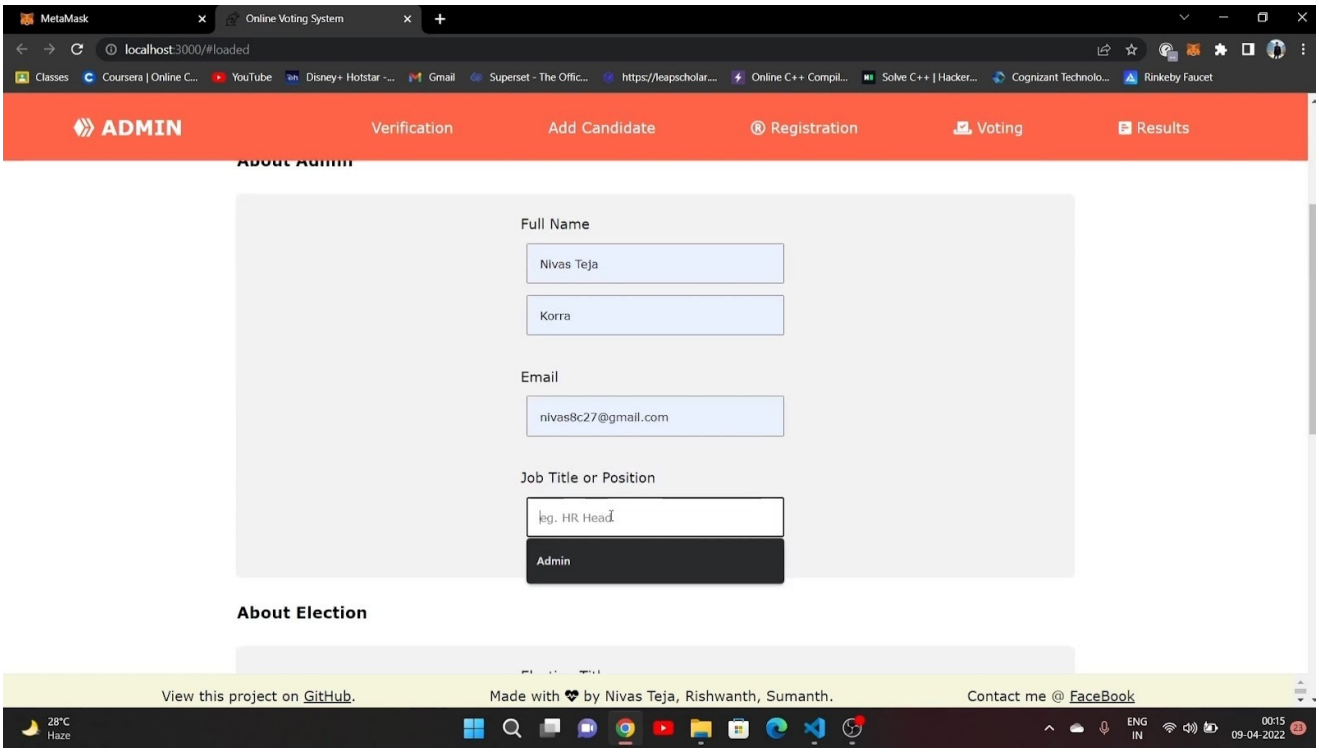


Figure 6.2: Admin Details

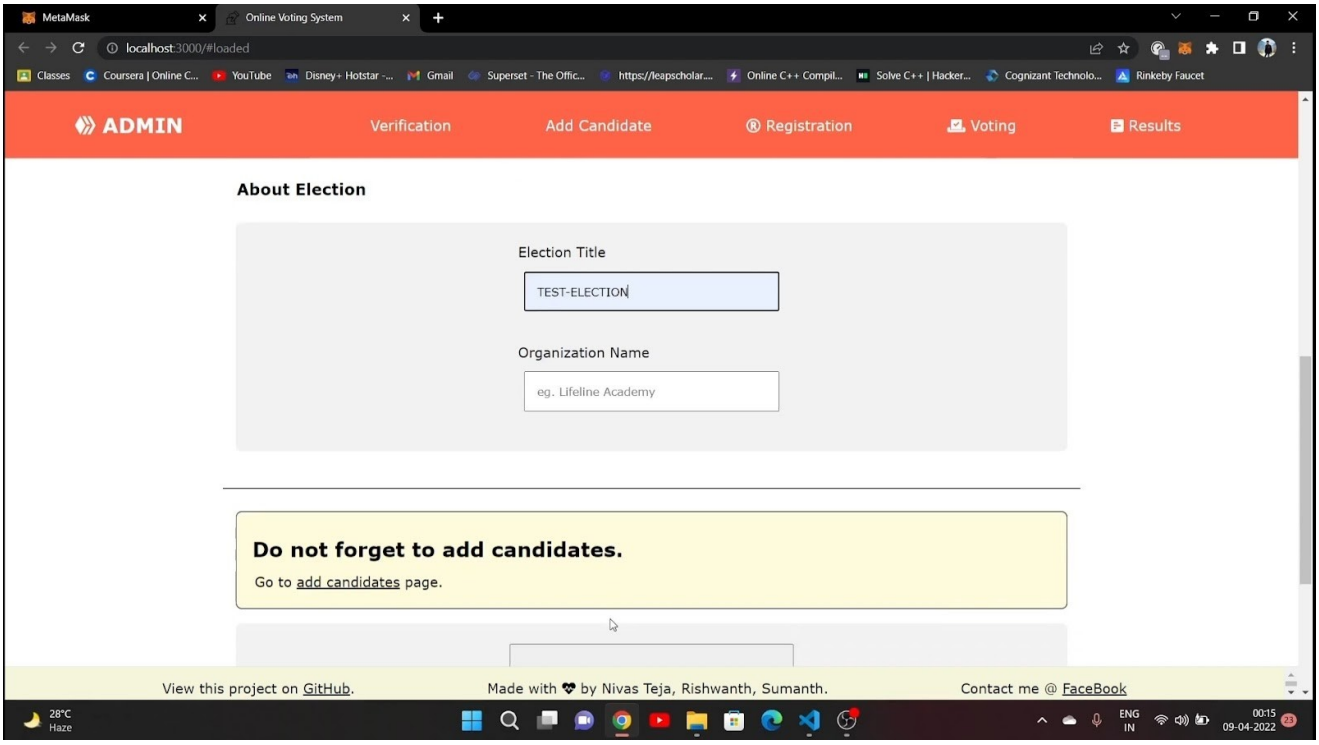


Figure 6.3: Election Creation

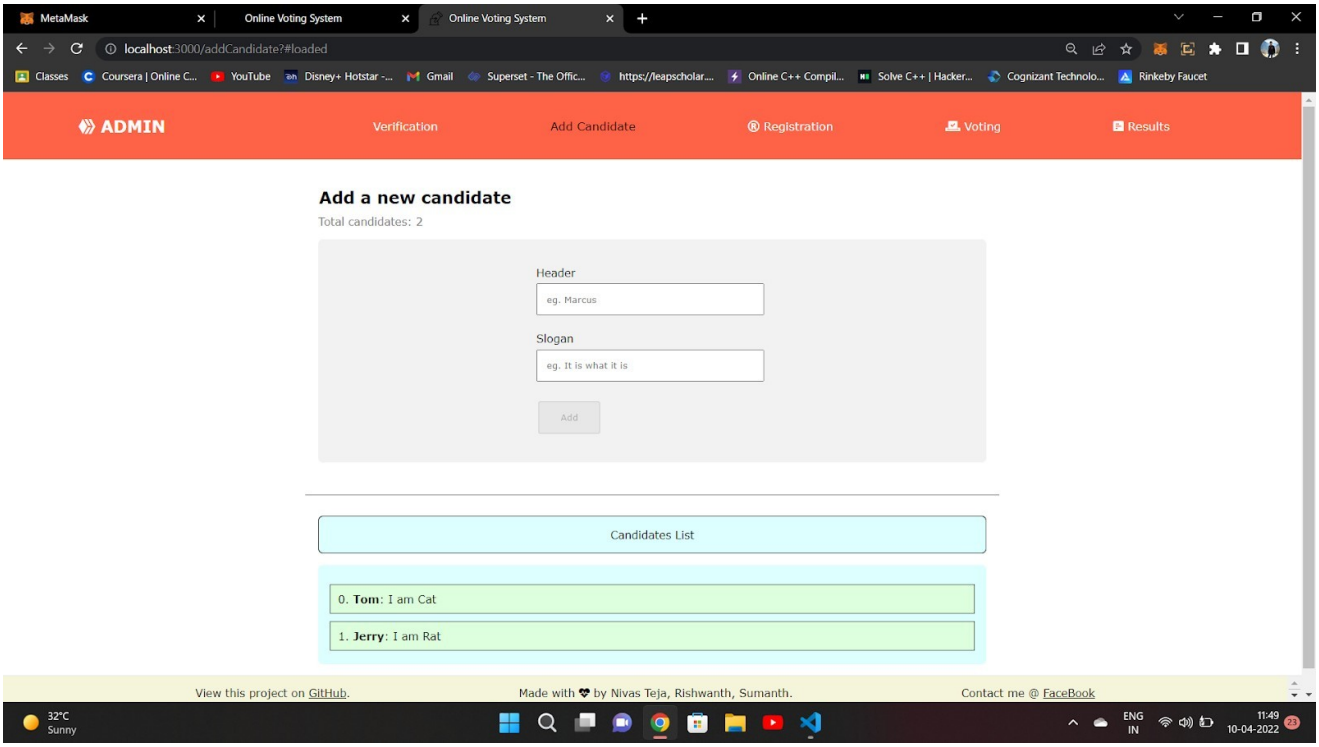


Figure 6.4: Add Candidate Page

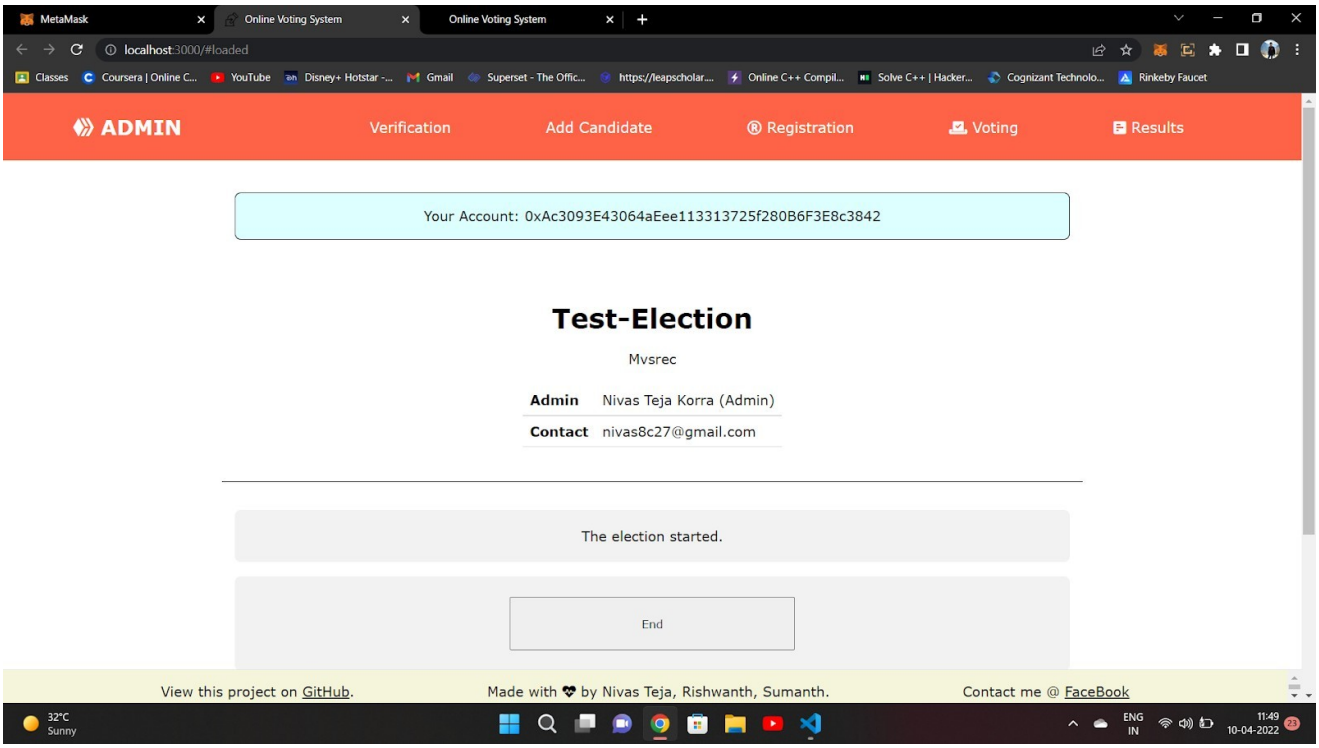


Figure 6.5: Admin Page

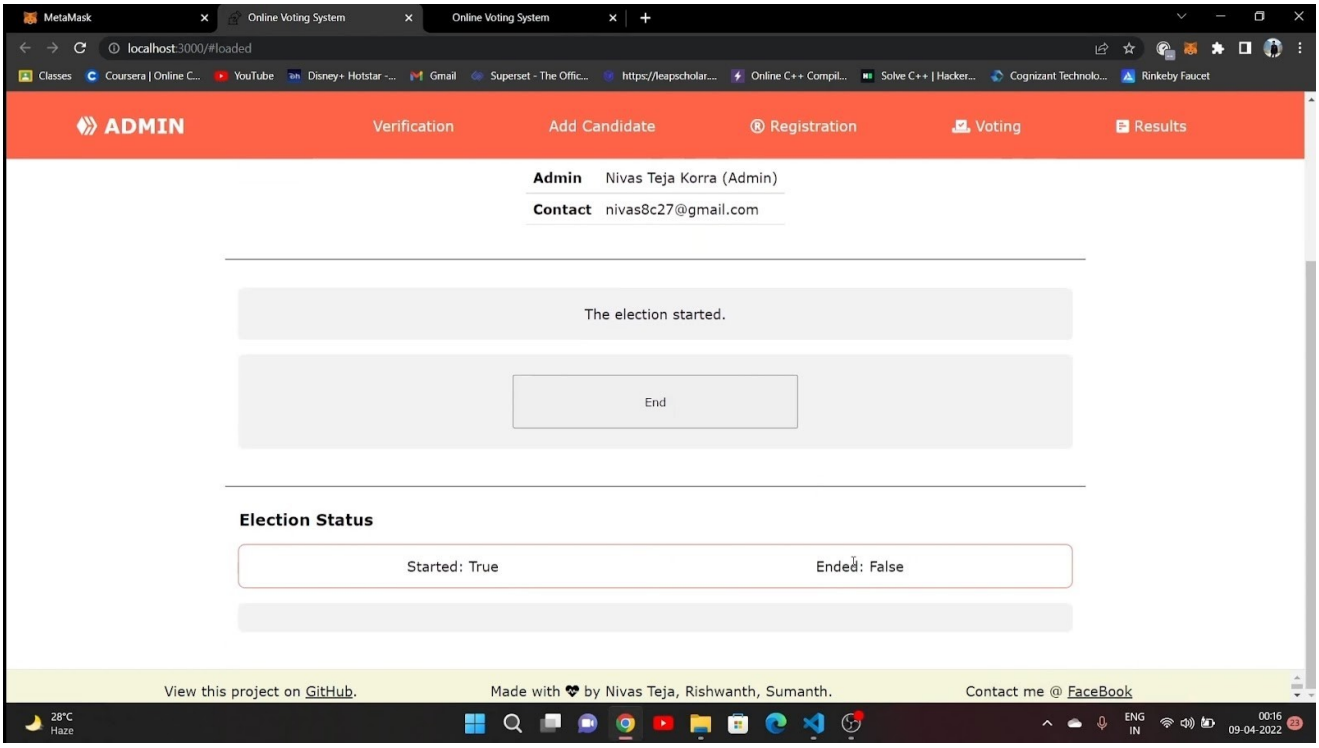


Figure 6.6: Election Status

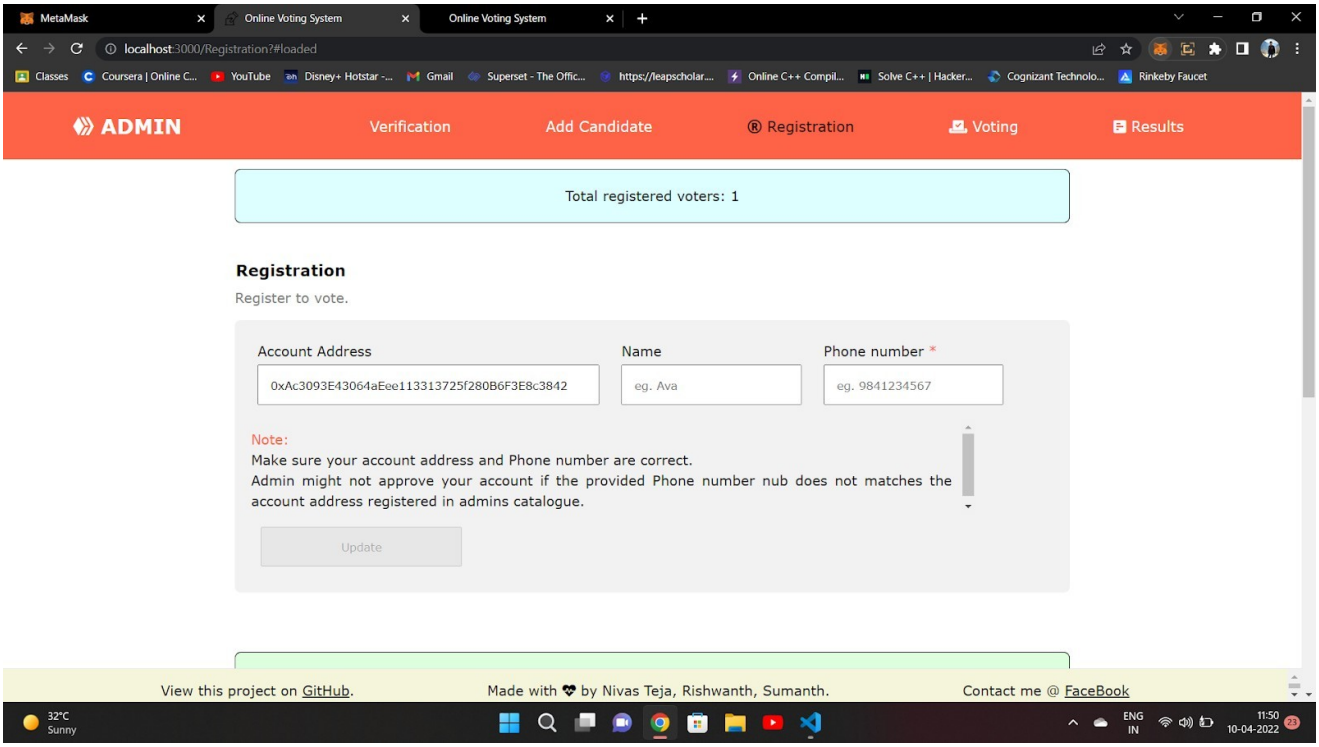


Figure 6.7: Voter Registration Page

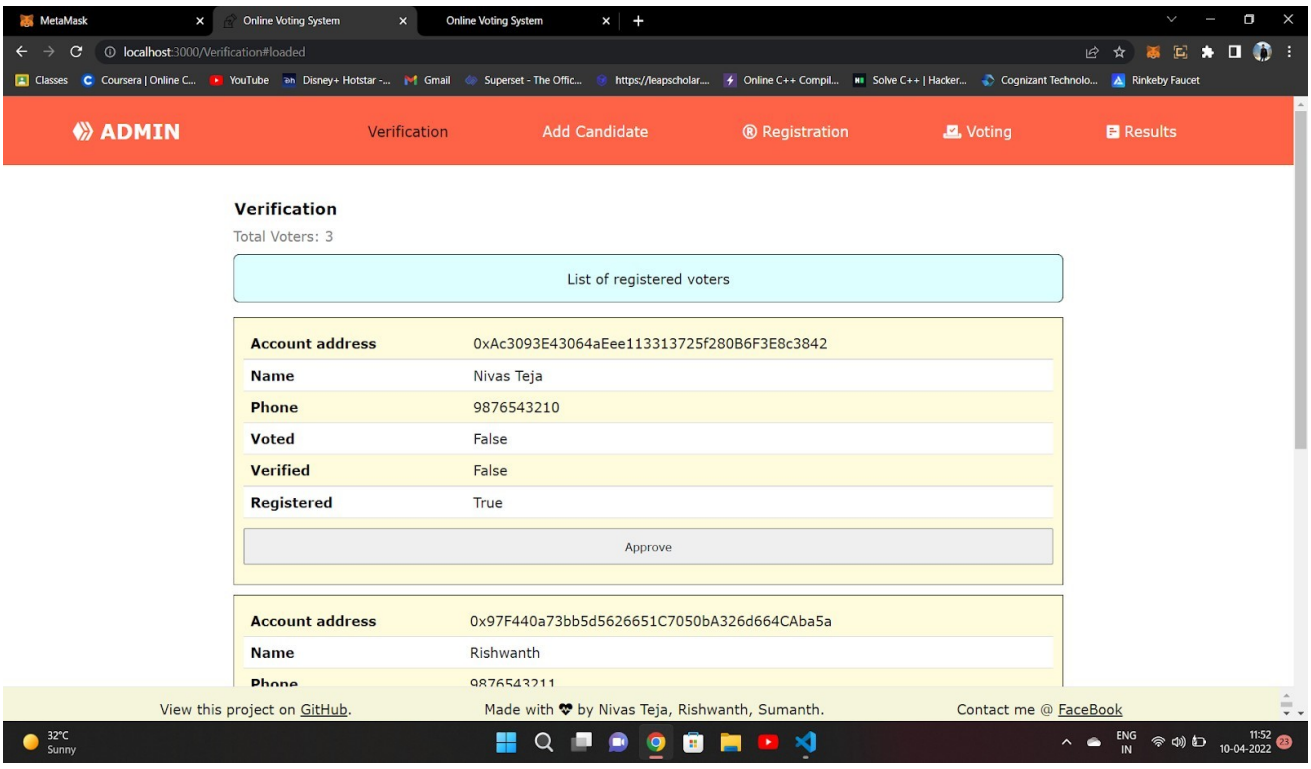


Figure 6.8: Before Verification

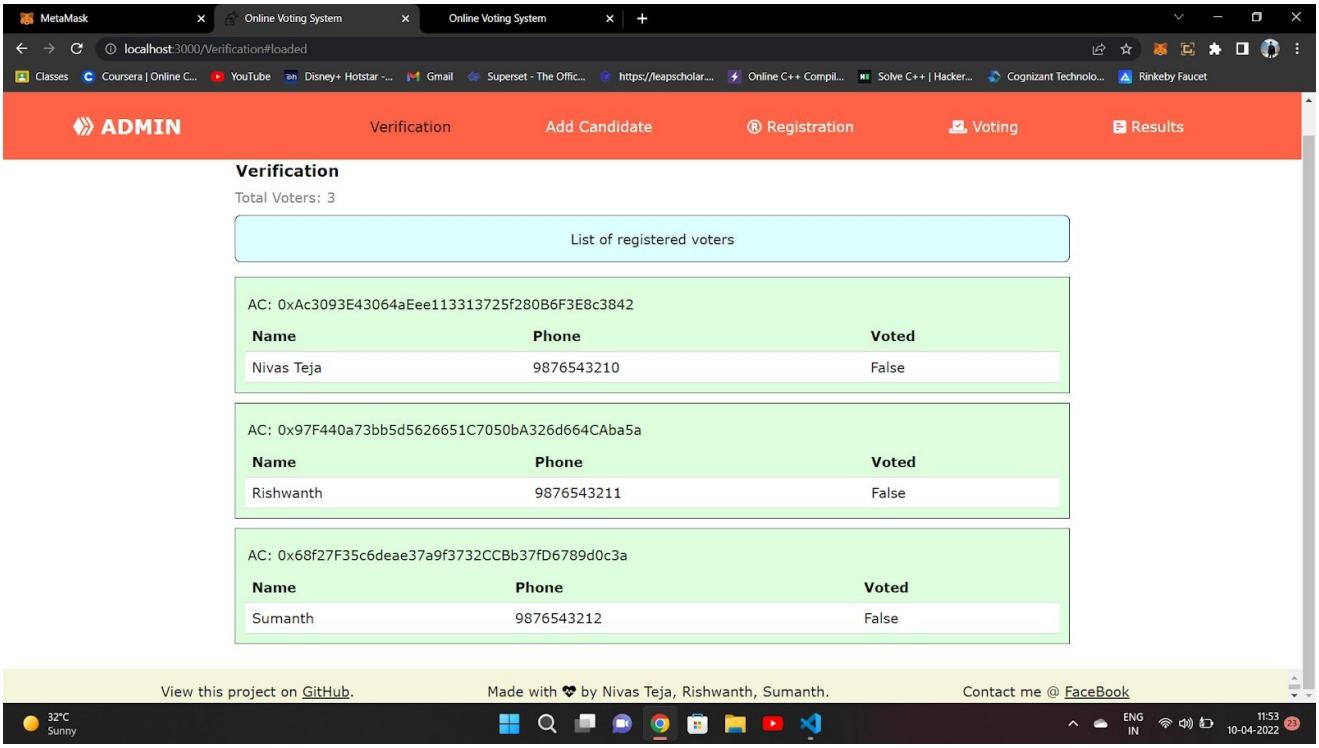


Figure 6.9: After Verification

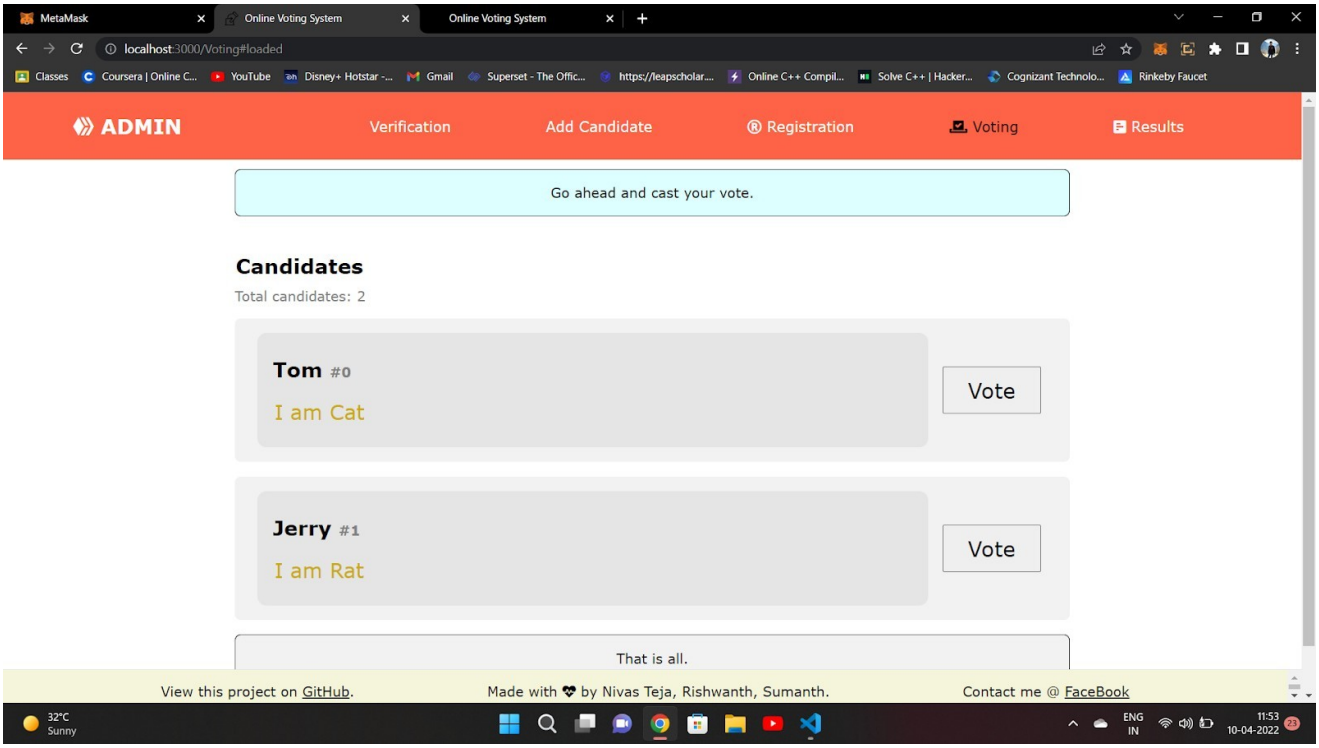


Figure 6.10: Voting Poll Page

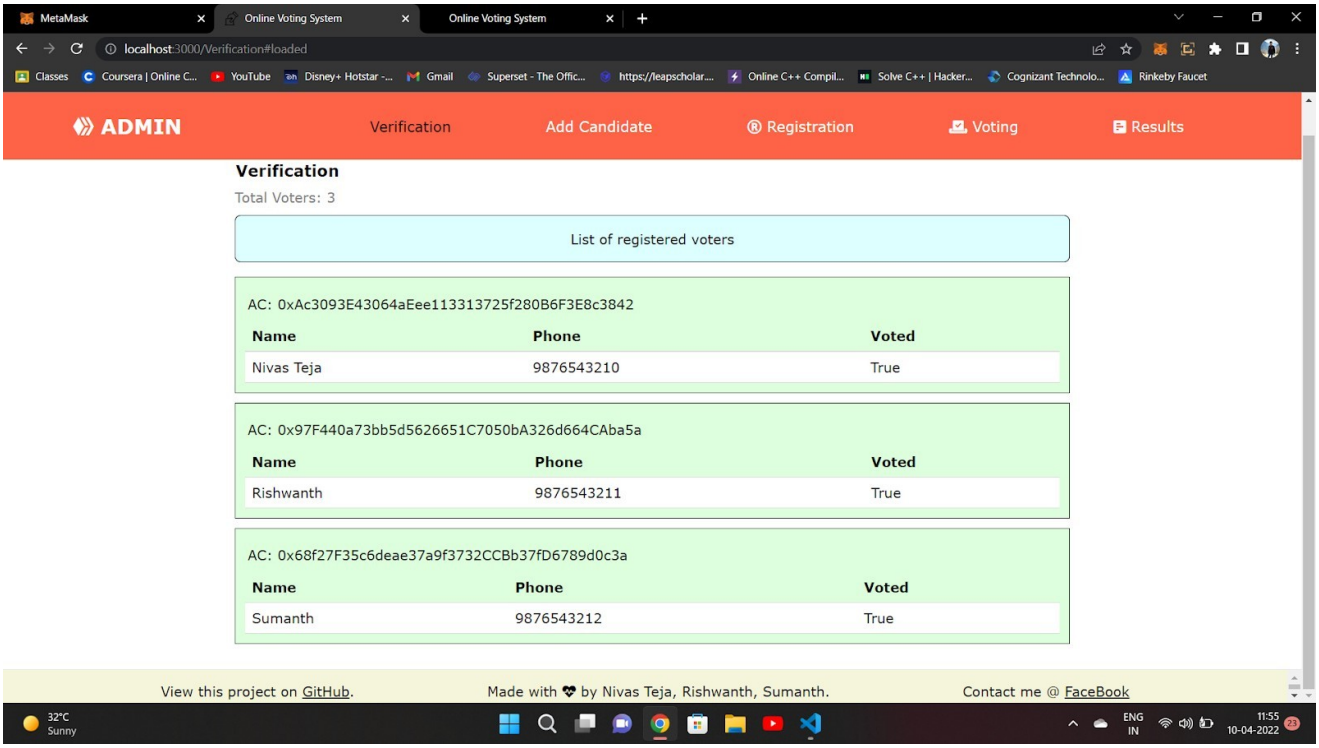


Figure 6.11: After Voting

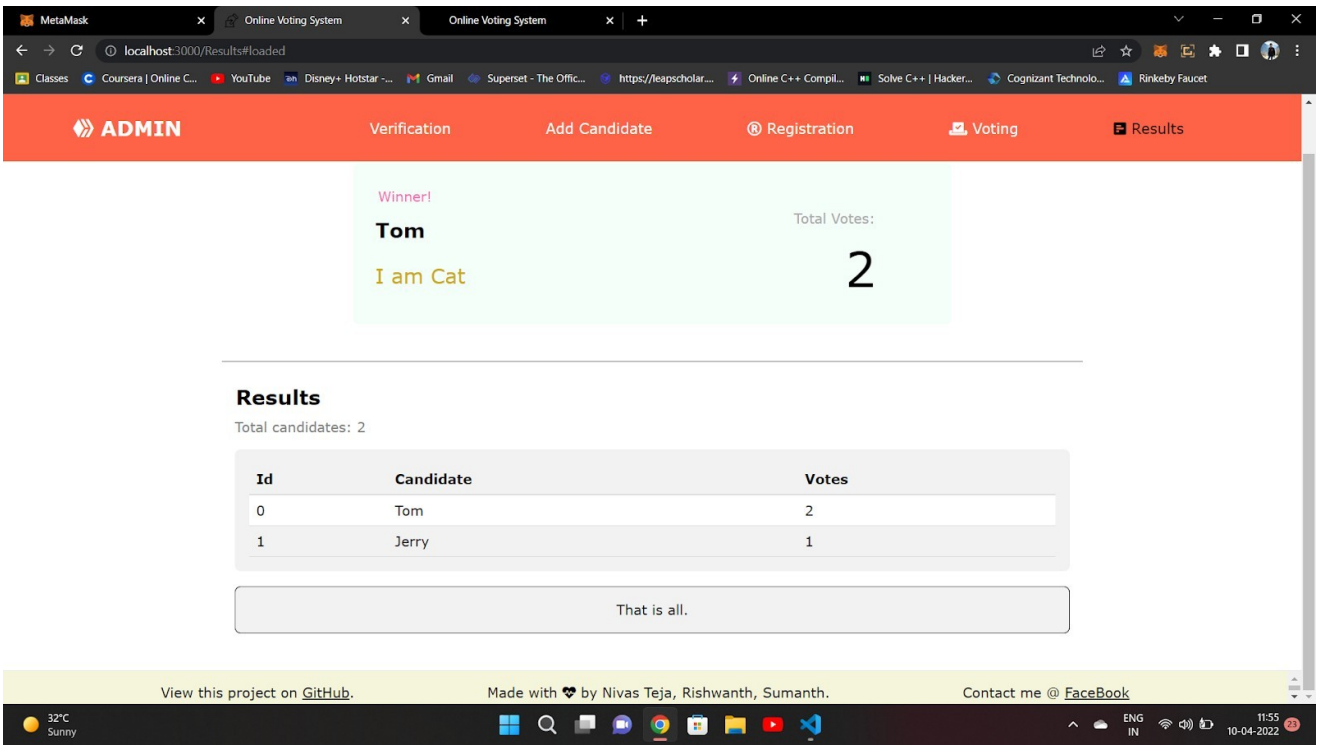


Figure 6.12 Result Page

CHAPTER 7

CONCLUSION

In this project we have developed an Online Voting System using Blockchain technology. It has overcome the problem of traditional voting systems, e-voting systems using blockchain is a promising research venture. Blockchain systems guarantee security, reliability, decentralized storage and anonymity. As a result, designing and implementing e-voting systems using blockchain ensures public and individual verifiability, dependability, reliability, consistency, auditability, anonymity, transparency, scalability, eligibility, authentication and fairness through principles of consensus, cryptography, digital signatures, and various blockchain mechanisms.

REFERENCES

[1] Rumeysa Bulut, Alperen Kantarcı, Safa Keskin, and Şerif Bahtiyar – "BlockchainBased Electronic Voting System for Elections in Turkey" in 2019 4th International Conference on Computer Science and Engineering (UBMK).

[2] Kirillov, Denis, Vladimir Korkhov, Vadim Petrunin, Mikhail Makarov, Ildar M. Khamitov, and Victor Dostov - "Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain" in ICCSA 2019: Computational Science and Its Applications.

[3] Veneta Aleksieva, Hristo Valchanov and Anton Huliyan – "Implementation of SmartContract, Based on Hyperledger Fabric Blockchain" in 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA).

[4] Dr Z. A Usmani, Kaif Patanwala, Mukesh Panigrahi, and Ajay Nair – "Multi-Purpose Platform Independent Online Voting System" in 2017 International Conference on Inovation in Information, Embedded and Communication Systems.

[5] Sunita Patil, Amish Bansal, Vaibhavi Pujari, Utkarsh Raina, and Roushan Kumar - "ESmart Voting System with Secure Data Identification using Cryptography" in 2018 3rd International Conference for Convergence in Technology (I2CT).

[6] Murat Kuzlu, Manisa Pipattanasomporn, Levent Gurses and Saifur Rahman – "Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability" in 2019 IEEE International Conference on Blockchain.

[7] Kriti Patidar and Dr. Swapnil Jain – "Decentralized E-Voting Portal Using Blockchain" in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[8] Ramya Govindaraj, Kumaresan P, and K.Sree harshitha – "Online Voting System using Cloud" in 2020 International Conference on Emerging Trends in Information Technology and Engineering.

APPENDIX

CODE:

```
pragma solidity >=0.4.21 <0.9.0;

contract Election {
    address public admin;
    uint256 candidateCount;
    uint256 voterCount;
    bool start;
    bool end;

    constructor() public {
        // Initilizing default values
        admin = msg.sender;
        candidateCount = 0;
        voterCount = 0;
        start = false;
        end = false;
    }

    function getAdmin() public view returns (address) {
        // Returns account address used to deploy contract (i.e. admin)
        return admin;
    }

    modifier onlyAdmin() {
        // Modifier for only admin access
        require(msg.sender == admin);
        _;
    }
    Modeling a candidate
    struct Candidate {
        uint256 candidateId;
        string header;
        string slogan;
        uint256 voteCount;
    }
    mapping(uint256 => Candidate) public candidateDetails;

    Adding new candidates
    function addCandidate(string memory _header, string memory _slogan)
        public
        // Only admin can add
        onlyAdmin
    {
        Candidate memory newCandidate =
            Candidate({
                candidateId: candidateCount,
                header: _header,
                slogan: _slogan,
                voteCount: 0
```

```
        });
    candidateDetails[candidateCount] = newCandidate;
    candidateCount += 1;
}
```

Modeling a Election Details

```
struct ElectionDetails {
    string adminName;
    string adminEmail;
    string  adminTitle;
    string electionTitle;
    string organizationTitle;
}
ElectionDetails electionDetails;

function setElectionDetails(
    string memory _adminName,
    string memory _adminEmail,
    string memory _adminTitle,
    string memory _electionTitle,
    string memory _organizationTitle
)
    public
    // Only admin can add
    onlyAdmin
{
    electionDetails = ElectionDetails(
        _adminName,
        _adminEmail,
        _adminTitle,
        _electionTitle,
        _organizationTitle
    );
    start = true;
    end = false;
}
```

Get Elections details

```
function getAdminName() public view returns (string memory) {
    return electionDetails.adminName;
}

function getAdminEmail() public view returns (string memory) {
    return electionDetails.adminEmail;
}

function getAdminTitle() public view returns (string memory) {
    return electionDetails.adminTitle;
}

function getElectionTitle() public view returns (string memory) {
    return electionDetails.electionTitle;
```

```
}

function getOrganizationTitle() public view returns (string memory) {
   return electionDetails.organizationTitle;
}
```

Get candidates count

```
function getTotalCandidate() public view returns (uint256) {
   // Returns total number of candidates
   return candidateCount;
}
```

Get voters count

```
function getTotalVoter() public view returns (uint256) {
   // Returns total number of voters
   return voterCount;
}
```

Modeling a voter

```
struct Voter {
   address voterAddress;
   string name;
   string  phone;
   bool isVerified;
   bool hasVoted;
   bool isRegistered;
}
address[] public voters; // Array of address to store address of voters
mapping(address => Voter) public voterDetails;
```

Request to be added as voter

```
function registerAsVoter(string memory _name, string memory _phone) public {
   Voter memory newVoter =
      Voter({
         voterAddress: msg.sender,
         name: _name,
         phone:    _phone,
         hasVoted:   false,
         isVerified:   false,
         isRegistered: true
      });
   voterDetails[msg.sender] = newVoter;
   voters.push(msg.sender);
   voterCount += 1;
}
```

Verify voter

```
function verifyVoter(bool _verifedStatus, address voterAddress)
   public
   // Only admin can verify
   onlyAdmin
{
```

```
    voterDetails[voterAddress].isVerified = _verifedStatus;
  }

  Vote
  function vote(uint256 candidateId) public {
    require(voterDetails[msg.sender].hasVoted == false);
    require(voterDetails[msg.sender].isVerified == true);
    require(start == true);
    require(end == false);
    candidateDetails[candidateId].voteCount += 1;
    voterDetails[msg.sender].hasVoted = true;
  }

  End election
  function endElection() public onlyAdmin {
    end = true;
    start = false;
  }

  Get election start and end values
  function getStart() public view returns (bool) {
    return start;
  }

  function getEnd() public view returns (bool) {
    return end;
  }
}
```

CO-PO/PSO MAPPING

Course code	**Statement** Student will be able to	**Cognitive Level**	**PO / PSO addressed**
PW961.1	Define a problem of the recent advancements with applications towards society.	An	PO1,PO2,PO3,PO4,PO5,PO6,PO7,PO8,PSO1,PSO2
PW961.2	Outline requirements and perform requirement analysis for solving the problem.	An	PO1,PO2,PO3,PO4,PO5,PO6,PO7,PO8,PSO1,PSO2
PW961.3	Design and develop a software and/or hardware based solution within the scope of project using contemporary technologies and tools.	AP, E, An	PO1,PO2,PO3,PO4,PO5,PO6,PO7,PO8,PSO1,PSO2
PW961.4	Test and deploy the applications for use.	AP,E,An	PO8,PO9,PO10,PO11,PO12,PSO1,PSO2
PW961.5	Develop the Project as a team and Demonstrate the application, with effective written and oral communications.	C	PO8,PO9,PO10,PO11,PO12,PSO1,PSO2

Table 1: Course Outcomes - Cognitive levels
Cognitive Levels: R-Remember; U-Understand; Ap-Apply; An=Analyze; E-Evluate; C-Create

Table 2: Number of performance indicators addressed by course outcomes

Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
No. of PIs addressed by course for a given PO	**4**	4	**4**	**4**	**4**	**4**	**4**	**5**	**5**	**7**	**4**	**6**	**5**	**6**
CO1	2	2	2	1	3	1	1	1	3	3	1	2	1	3
CO2	3	2	3	2	3	1	1	1	3	3	3	2	2	3
CO3	3	3	3	2	3	1	1	3	3	3	3	2	2	3
CO4								3	3	3	3	2	2	2
CO5								1	3	3	3	2	2	3

Table 3: Calculation of CO-PO/PSO correlation levels

	PO1		PO2		PO3		PO4		PO5		PO6		PO7		PO8		PO9		PO10		PO11		PO12		PSO1		PSO2	
PW961IT	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level	%	Level
CO1	50	3	50	2	50	2	25	1	50	3	25	1	25	1	20	1	50	3	50	3	25	1	30	1	20	1	50	3
CO2	75	3	50	2	75	3	50	2	50	3	25	1	25	1	20	1	50	3	50	3	75	3	30	1	50	3	50	3
CO3	75	3	75	3	75	3	50	2	50	3	25	1	25	1	50	3	50	3	75	3	75	3	30	1	50	3	50	3
CO4	50														50	3	50	3	75	3	75	3	30	1	20	1	50	3
CO5	50														20	1	50	3	50	3	75	3	30	1	40	2	50	3
No. Mapped	3	9	3	7	3	11	3	8	3	9	3	3	3	3	5	9	5	15	5	15	5	13	5	5	5	10	5	15
Average of Level	9/3=3		7/3=2.3		8/3=2.6		8/3=2.6		9/3=3		3/3=1		3/3=1		9/5=1.8		15/5=3		15/5=3		13/5=2.6		5/5=1		10/5=2		15/5=3	
Rounded average level	**3**		**2**		**3**		**3**		**3**		**1**		**1**		**2**		**3**		**3**		**3**		**1**		**2**		**3**	

Table 4: Course Articulation Matrix

PW961IT	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	3	1	1	1	3	3	1	1	1	3
CO2	3	2	3	2	3	1	1	1	3	3	3	1	3	3
CO3	3	3	3	2	3	1	1	3	3	3	3	1	3	3
CO4								3	3	3	3	1	1	3
CO5								1	3	3	3	1	2	3
PW961IT	**3**	**2**	**3**	**3**	**3**	**1**	**1**	**2**	**3**	**3**	**3**	**1**	**2**	**3**

RUBRICS ANALYSIS

Table 5: PO/PSO addressed by the Project

Project Name	Domain	In-house / Industry	PO/PSO addressed	Name and Signature of Guide

Table 6: Rubrics Evaluation

PO/PSO	PO1,PO2,PO6,PO7				PO3	PO4,PO5, PSO1	PO4,PO5, PSO2	PO8	PO9				PO10			PO11	PO12
Rubrics	R1				R2	R3	R4	R5	R6				R7			R8	R9
Roll. No.	CI	CII	CIII	Total	CIV	CV	CVI	CVII	CVIII	CIX	CX	Total	CXI	CXII	Total	CXIII	CIV
	4	4	4	12	4	4	4	4	4	4	4	12	4	4	8	4	4

Rubrics for project

Focus Areas:

1. **Problem Formulation (PO1,PO2, PO6,PO7)**
2. **Project Design (PO3)**
3. **Build (PO4,PO5,PSO1)**
4. **Test& Deploy (PO4, PO5,PSO2)**
5. **Ethicalresponsibility (PO8)**
6. **Team Skills (PO9)**
7. **ProjectPresentation (P10)**
8. **Projectmanagement (PO11)**
9. **LifelongLearning (PO12)**

Focus Areas	Criterion [c]	Exemplary 4	Satisfactory 3	Developing 2	Unsatisfactory 1
Problem Formulation (PO1,PO2, PO6, PO7)	**I - Identify/Define Problem** Ability to identify a suitable problem and define the project objectives.	Demonstrates a skillful ability to identify / articulate a problem and the objectives are well defined and prioritized.	Demonstrates ability to Identify / articulate a problem and All major objectives are identified.	Demonstrates some ability to identify / articulate a problem that is partially connected to the issues and most major objectives are identified but one or two minor ones are missing orpriorities are not established.	Demonstrates minimal or no ability to identify / articulate a problem and many major objectives are not identified.
	II - Collection of Background Information: Ability to gather background Information (existing knowledge, research, and/or indications of the problem)	Collects sufficient relevant background information from appropriate sources, and is able toidentify pertinant/critical information;	Collects sufficient relevant background information from appropriatesources;	Collects some relevant background information from appropriate Sources.	Minimal or no ability to collect relevant backgroundinformation
	III- Define scope of the problem Ability to identify problem scope suitable to the degree considering the impact on society and environment	Demonstrates a skillful ability to define the scope of problem accurately mentioning the relevant fields of engineering precisely. Considers, explainsand evaluates the impact of engineering interventions on society and environment.	Demonstrates ability to define problem scope mentioning the relevant fields of engineering broadly. Considers and explains the impact of engineering interventions on society and environment	Demonstrates some ability todefine problem scope mentioning some of the relevant fields. Some consideration of the impact of engineering interventions on society and environment.	Demonstrates minimal or no ability to define problem scope and fails to mention relevant fields of engineering. Minimal or no consideration of the impact of engineering interventions on society and environment
Project Design (PO3)	**IV- Understandingthe Design Process and Problem Solving:** Ability to explain the design process including the importance of needs, specifications, concept generation and to develop an approachto solve a problem.	Demonstrates a comprehensive ability to understand and explain a design process. Considers multiple approaches to solving a problem, and can articulate reason for choosing solution	Demonstrates an ability to understand and explain a design process. Considers multiple approaches to solving a problem, which is justified and considers consequences.	Demonstrates some ability to understand and explain a design process. Considers a few approaches to solving a problem; doesn't always consider consequences.	Demonstrates minimal or no ability to understand and explain a design process. Considers a single approach to solving a problem. Does not considerconsequences.

Build (PO4,PO5, PSO1)	**V- Implementing Design Strategy:** Ability to execute a solution taking into consideration design requirements using appropriate tool (software/hardware);	Demonstrates a skillful ability to execute a solution taking into consideration all design requirements using the most relevant tool.	Demonstratesanabilityto execute a solution taking into consideration design requirements using relevanttool.	Demonstrates some ability to execute a solution but not using most relevant tool.	Demonstrates minimal or no ability to execute a solution. Solution does not directly attend to the problem.
Test & Deploy (PO4, PO5, PSO2)	**VI- Evaluating Final Design:** To evaluate/confirm the functioning of the final design. To deploy the project on the target environment	Demonstrates a skillful ability to evaluate/confirm the functioning of the final design skillfully, with deliberation for further Improvement after deployment.	Demonstrates an ability to evaluate/confirm the functioning of the final design. The evaluation is complete and has sufficient depth.	Ability to evaluate/confirm the functioning of the final design, but the evaluation lacks depth and/or is incomplete.	Demonstrates minimal or no ability to evaluate/confirm the functioning of the final design.
Ethical responsibility (PO8)	**VII - Proper Use of Others' Work:** Ability to recognize, understand and apply proper ethical use of intellectual property, copyrightedmaterials, and research.	Always recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Some recognition and application of proper ethical use ofintellectual property, copyrighted materials, and others' research.	Minimal or no recognition and/or application of proper ethical use of intellectual property, Copyrighted materials, or others'research.
Team Skills (PO9)	**VIII - Individual Work Contributions and Time Management:** Ability to carry out individual Responsibilities and manage time (estimate, prioritize, establish deadlines/ milestones, follow timeline, plan for contingencies, adapt to change).	Designated jobs are accomplished by deadline; completed work is carefully and meticulously prepared and meets allrequirements.	Designated jobs are accomplished by deadline; completed work meets requirements.	Designated jobs are accomplished by deadline; completed work meets most requirements.	Some Designated jobs are accomplished by deadline; completed work meetssome requirements.
	IX - Leadership Skills: Ability to lead a team. (i) Mentors and accepts mentoring fromothers. (ii) Demonstrates capacity for initiative while respecting others' roles. (iii) Facilitates others' involvment. (iv) Evaluates team Effectiveness andplans for improvements	Exemplifies leadership skills.	Demonstrates leadership skills.	Demonstrates some leadership skills at times.	Demonstrates minimal or no Leadership skills.
	X - Working with Others: Ability to listen to, collaborate with, and champion the efforts of others.	Skillfully listens to, collaborates with, and champions the efforts of others.	Listens to, collaborates with, and champions the efforts of others.	Sometimes listens to, collaborates with, and champions others' efforts.	Rarely listens to, collaborates with, or champions others' efforts.

Project Presentation (P10)	**XI - Technical Writing Skills** Ability to communicate the main idea with clarity. Ability to use illustrations properly to support ideas (citations, position on pageetc)	Main idea is clearly and precisely stated. Materials are seamlessly arranged in a logical sequence Illustrations are skillfully used to supportideas	Main idea is understandable. Material moves logically forward, Illustrations are properly used to supportideas	Main idea is somewhat Understandable. Material has some logical order and is somewhat coherent or easy to follow. Illustrations are for the most part properly used to support ideas	Main idea is difficult to understand. Material has little logical order, and is often unclear, incoherent. Illustrations are used, but minimally support ideas. (not properly cited etc)
	XII - Communication Skills for Oral Reports Ability to present strong key ideas and supporting details with clarity andconcision. Maintain contact with audience, and ability to complete in the allotted time	Presentation logically and skillfully structured. Key ideas are compelling, andarticulated with exceptional clarity and concision. Introduction, supporting details and summary are clearly evident and memorable, and ascertain the credibility of the speaker Presentation fits perfectly within timeconstraint.	Presentation has clear structure and is easy to follow. Key ideas are clearly and concisely articulated, andare interesting. There is sufficient detail to ascertain speaker's authority, and presentation includes an introduction and summary. Presentation fits within time constraint, though presenter might have to subtly rush or slow down.	Presentation has some structure. Key ideas generally identifiable, although not very remarkable. Introduction, supporting details and/or summary may be too broad, too detailed or missing. Credibility of the speaker may be questionable at times. Presentation does not quite fit within time constraint; presenter has to rush or slow down at end	Presentation rambles. Not organized; key ideas are difficult to identify, and are unremarkable. No clear introduction, supporting details and summary. Speaker has no credibility. Presentation is unsuitablably short or unreasonablylong.
Project management (PO11)	**XIII - Monitoring and Controlling the Project**	Monitors timelines and progress toward project goals on a daily basis. Provides accurate, complete reports of project progress.	Monitors timelines and progress toward project goals most of the time. Provides relatively accurate, complete reports of project progress with only mi nor errors or omissions	Seldom monitors timelines and progress toward project goals. Provides relatively acc urate, yet clearly incomplete, reports of project progress	Does not monitor timeline s and progress toward project goals. Provides inaccurate, incomplete reports of project progress
Lifelong Learning (PO12)	**XI V - Extend Scope of Work:** Ability to extend the project through implementation in other studyareas	Demonstrates a skillful ability to explore a subject/topic thoroughly, discusses the road map to extend the project in otherareas.	Demonstrates an ability to explore a subject/topic, and shows possible areas in which project can be extended	Demonstrates some ability to explore a subject/topic, providing some knowledge of areas in which project can beextended	Demonstrates minimal or no ability to explore a subject/topic, and does not discuss future work clearly mentioning other areas