A

Mini Project

On

A BLOCK CHAIN E-VOTING

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

MD VAHEED (197R1A0591)
MUDASSAR AHMED KHAN (197R1A0593)
MD ABDUL RAHMAN (207R5A0511)

Under the Guidance of

K. PRAVEEN KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road,

Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**A BLOCK CHAIN E - VOTING" being** submitted by **MD VAHEED (197R1A0591) , MUDASSAR AHMED KHAN (197R1A0593) , MD ABDUL RAHMAN (207R5A0511)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

K. PRAVEEN KUMAR
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on_____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **K. PRAVEEN KUMAR,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Mr. G. Vijay Kumar, Dr. M. Subha Mastan Rao & J. Narasimha Rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages. We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project. The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

MD VAHEED (197R1A0591)

MUDASSAR AHMED KHAN (197R1A0593)

MD ABDUL RAHMAN (207R5A0511)

ABSTRACT

Some forms of voting have been here ever since. Mostly used form all over the world are paper ballots. Electronic voting schemes are being popular only in the last decade and they are still unsolved. E-voting schemes bring problems mainly regarding security, credibility, transparency, reliability, and functionality. Estonia is the pioneer in this field and may be considered the state of the art. But there are only a few solutions using blockchain. Blockchain can deliver an answer to all of the mentioned problems and furthermore bring some advantages such as immutability and decentralization. The main problems of technologies utilizing blockchain for e-voting are their focus on only one field or lack of testing and comparison. In this paper, we present a blockchain-based e-voting platform, which can be used for any kind of voting. It is fully utilized by blockchain and all processes can be handled within it. After the start of the voting, the platform behaves as fully independent and decentralized without possibilities to affect the voting process. The data are fully transparent, but the identity of voters is secured by homomorphic encryption. We have tested and compared our solution in three different blockchains. The results show that both public and private blockchains can be used with only a little difference in the speed. The key novelty of our solution is a fully decentralized management of e-voting platform through blockchain, transparency of the whole process and at the same time security and privacy of the voters thanks to homomorphic encryption.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled “A BLOCK CHAIN E-VOTING”. by name we can understand that Online voting is a trend that is gaining momentum and trend in modern society. The main goal of this analysis was to examine the current status of blockchain-based voting research and online voting systems and any related difficulties to predict future developments.

1.2 PROJECT PURPOSE

This study provides a conceptual description of the intended blockchain-based electronic voting application and an introduction to the fundamental structure and characteristics of the blockchain in connection to electronic voting.

1.3 PROJECT FEATURES

The main features of this project is it has great potential to decrease organisational costs and increase voter turnout. It eliminates the need to print ballot papers or open polling stations—voters can vote from wherever there is an Internet connection

Key features of Blockchain Project:

- ❖ High Availability
- ❖ Verifiability
- ❖ Transparency
- ❖ Immutability
- ❖ Decentralised
- ❖ Enhanced Security

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

Systems analysis the process of observing systems for troubleshooting or development purposes. It is applied to information technology, where computer-based systems require defined analysis according to their makeup and design. systems analysis can include looking at end-user implementation of a software package or product; looking in-depth at source code to define the methodologies used in building software; or taking feasibility studies and other types of research to support the use and production of a software product, among other things.

2.1 PROBLEM DEFINITION

Blockchain era fixed shortcomings in these day's method in elections made the polling mechanism clean and accessible, stopped unlawful vote casting, bolstered the statistics safety, and checked the final results of the polling. but electronic vote casting includes widespread risks such as if an electronic voting system is compromised, all forged votes can probable be manipulated and misused. digital vote casting has for that reason now not yet been adopted on a country wide scale, thinking about all its possible benefits. these days, there's a viable solution to conquer the dangers and electronic voting, that's blockchain technology.

2.2 EXISTING SYSTEM

In recent years blockchain is frequently mentioned for example of relaxed era utilized in online surroundings. Our e-voting system makes use of blockchain to control all election techniques. Its fundamental benefit is that there's no want for self-belief within the centralized authority that created the elections. This authority can't have an effect on the election effects in our machine. another task in e-balloting is the shortage of transparency inside the functioning of the gadget, leading to a lack of self-belief in electorate.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- It is a manual process.
- It is a new growing technology hence information and awareness is much needed.

2.3 PROPOSED SYSTEM

The proposed blockchain voting system considers all requirements for voting and is designed generally for any elections e.g., president, student parliament, etc. The system allows more round elections and preferably uses a public blockchain. The public blockchain can be replaced by other types of blockchain but the stored data (votes) have to be easily verified by any user. The user represents any observer who is interested in the blockchain voting. In our proposed system we identify three main roles: vote publisher; key authority; and voter. These three roles can represent an organization, a company, or a user. The roles vote publisher and key authority can be grouped to one role due to that they can be the same organization or person. The voter attends the elections depending on vote configuration. The configuration of the votes is performed by the vote publisher and is included in the smart contract. The vote publisher has to know all cipher keys before publishing the smart contract. The close collaboration between the vote publisher and the key authority is required. The key authority creates and distributes all cipher keys to a voter and vote publisher. The distributing channel has to be secured and should not be vulnerable to any 3rd party.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- We can vote through automatic process.
- It is easy to vote.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the

proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and

to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- System : intel core i3.
- Hard Disk : 256 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB OR ABOVE

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 10
- Coding Language : python
- Tool : PyCharm
- Database : MYSQL
- Server : Flask

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for Block Chain e-voting:

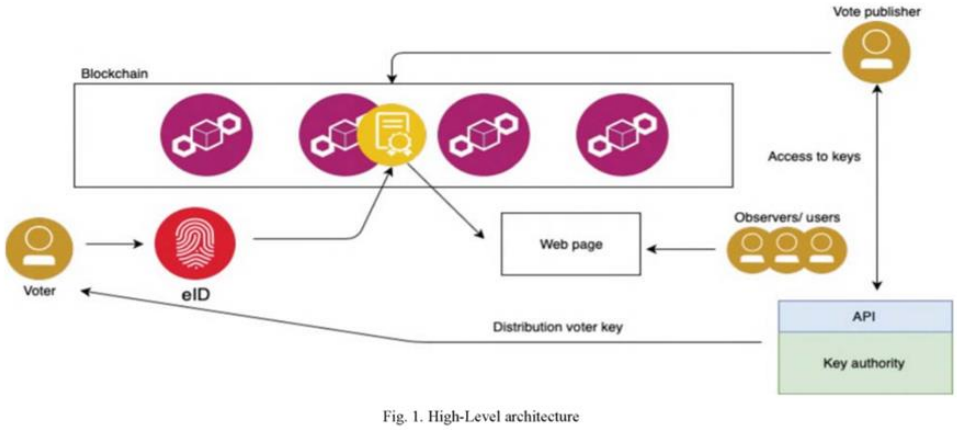


Fig. 1. High-Level architecture

Figure 3.1: Project Architecture of Block Chain e-voting

3.2 DESCRIPTION

In this project we are using public python Blockchain API's to store and manage voting data as Blockchain provides secure and tamper proof of data storage and to implement this project we have designed following modules.

Admin module: this user responsible to add new party and candidate details and can view party details and vote count. Admin login to system by using username as 'admin' and password as 'admin'.

User Module: this user has to sign up with the application by using username as his ID and then upload his face photo which capture from webcam. After registering user can go for login which validate user id and after successful login user can go for cast vote module which execute following functionality

3.3 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

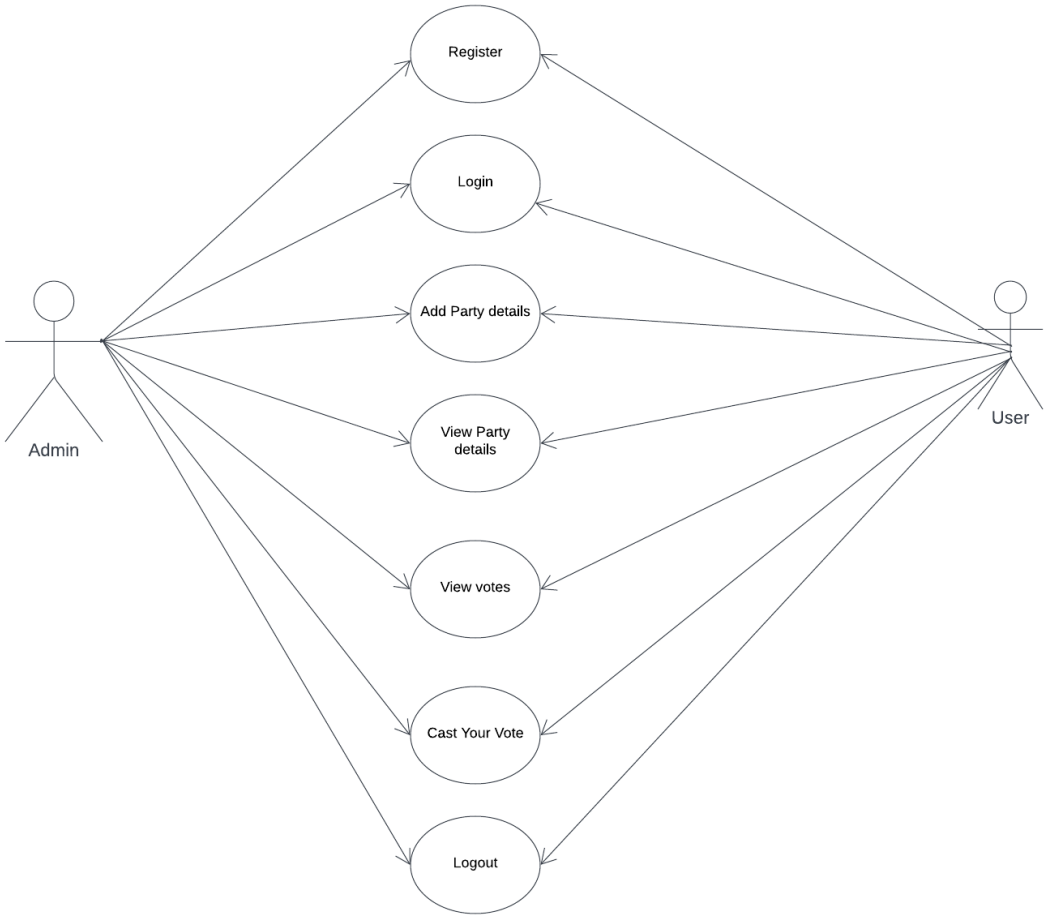


Figure 3.3: Use Case Diagram for Block Chain e-voting web portal model

3.4 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 information.

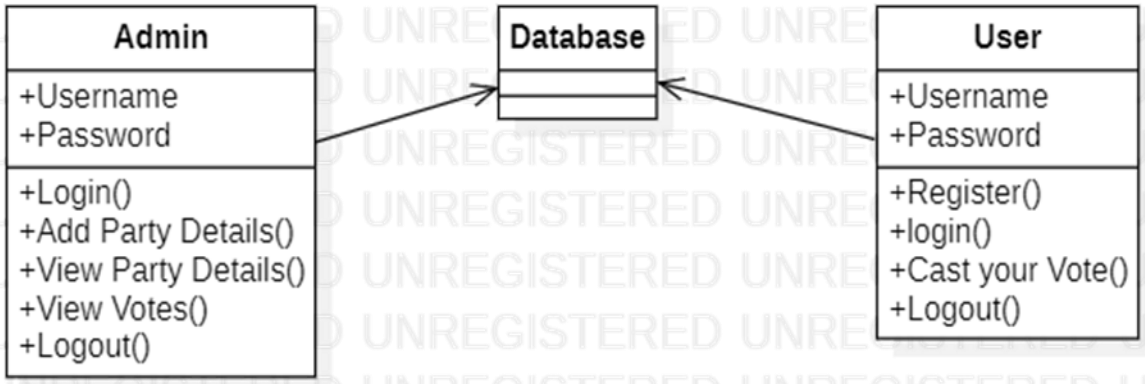


Figure 3.4: Class Diagram for Block Chain e-voting Application

3.5 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.

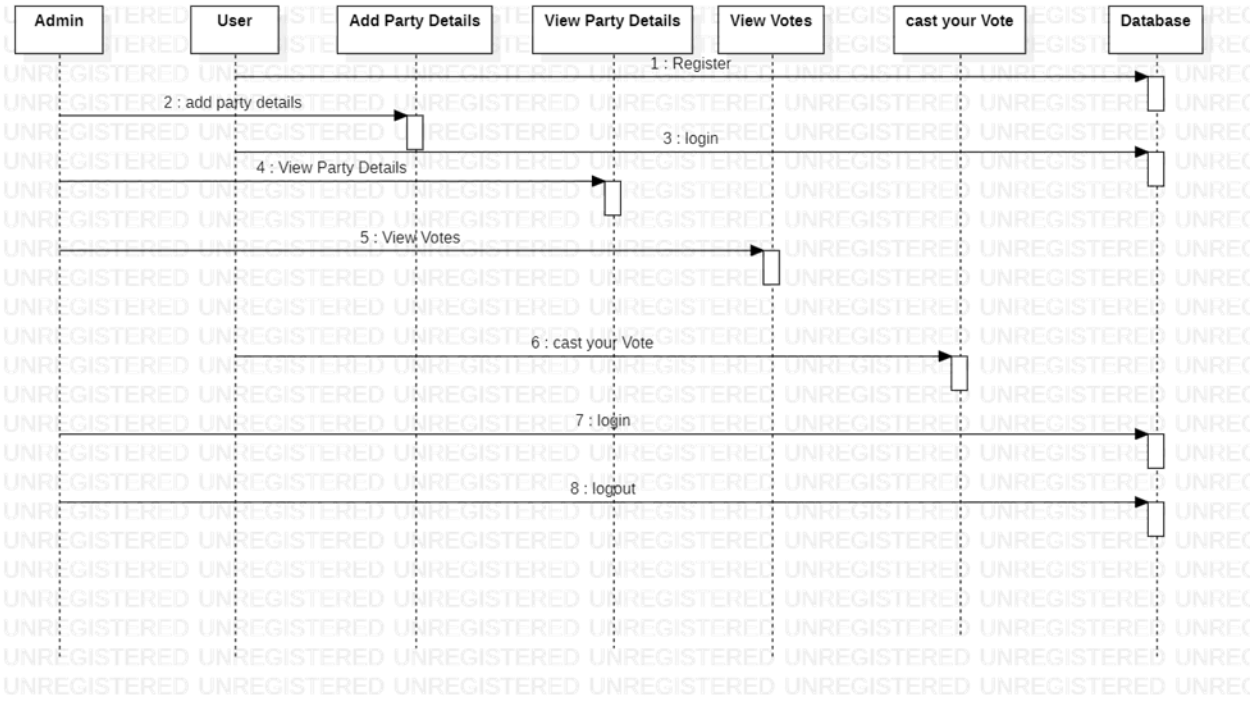


Figure 3.5: Sequence Diagram for Block Chain e-voting

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

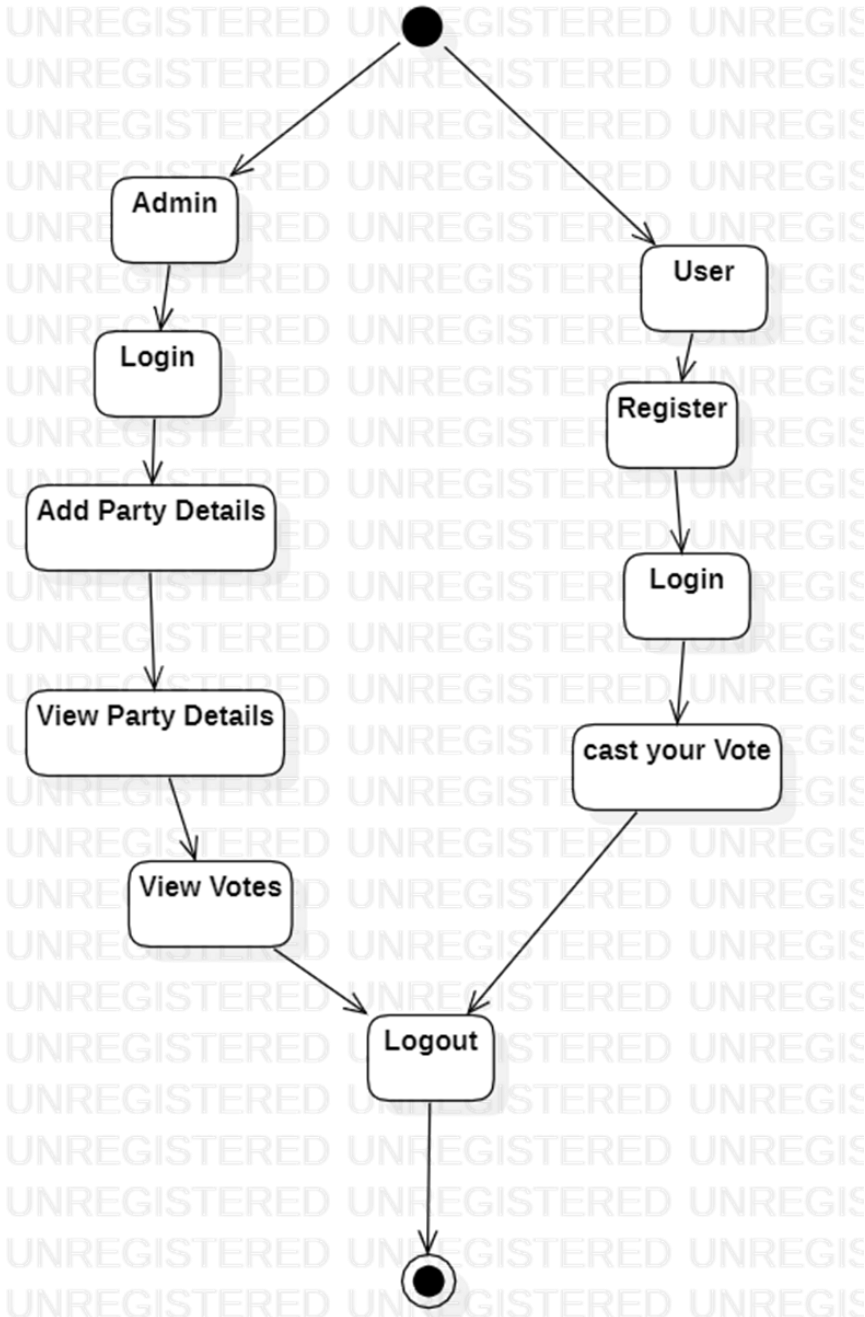


Figure 3.6: Activity Diagram for A Block Chain e-voting

4. IMPLEMENTATION

4.1 SAMPLE CODE

Blockchain.py

```
from hashlib import sha256
import json
import time
import pickle
from datetime import datetime
import random
import pyaes, pbkdf2, binascii, os, secrets
import base64


class Block:
    def __init__(self, index, transactions, timestamp, previous_hash):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = 0


    def compute_hash(self):
        block_string = json.dumps(self.__dict__, sort_keys=True)
        return sha256(block_string.encode()).hexdigest()


class Blockchain:
    # difficulty of our PoW algorithm
```

```
    difficulty = 2 #using difficulty 2 computation
    def __init__(self):
        self.unconfirmed_transactions = []
        self.chain = []
        self.create_genesis_block()
        self.peer = []
        self.translist = []

    def create_genesis_block(self): #create genesis block
        genesis_block = Block(0, [], time.time(), "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        return self.chain[-1]

    def add_block(self, block, proof): #adding data to block by computing new and previous hashes
        previous_hash = self.last_block.hash

        if previous_hash != block.previous_hash:
            return False
        if not self.is_valid_proof(block, proof):
            return False
```

```
        block.hash = proof
            #print("main "+str(block.hash))
            self.chain.append(block)
            return True
        def is_valid_proof(self, block, block_hash): #proof of work
            return (block_hash.startswith('0' * Blockchain.difficulty) and block_hash == block.compute_hash())

        def proof_of_work(self, block): #proof of work
            block.nonce = 0

            computed_hash = block.compute_hash()
            while not computed_hash.startswith('0' * Blockchain.difficulty):
                block.nonce += 1
                computed_hash = block.compute_hash()

            return computed_hash

        def add_new_transaction(self, transaction):
            self.unconfirmed_transactions.append(transaction)

        def addPeer(self, peer_details):
            self.peer.append(peer_details)

        def addTransaction(self,trans_details): #add transaction
            self.translist.append(trans_details)
```

```
    def mine(self):#mine transaction
        if not self.unconfirmed_transactions:
            return False

        last_block = self.last_block

        new_block = Block(index=last_block.index + 1,
                          transactions=self.unconfirmed_transactions,
                          timestamp=time.time(),
                          previous_hash=last_block.hash)

        proof = self.proof_of_work(new_block)
        self.add_block(new_block, proof)

        self.unconfirmed_transactions = []
        return new_block.index

    def save_object(self,obj, filename):
        with open(filename, 'wb') as output:
            pickle.dump(obj, output, pickle.HIGHEST_PROTOCOL)
```

Manage.py

```
#!/usr/bin/env python
import os
import sys
if __name__ == '__main__':
```

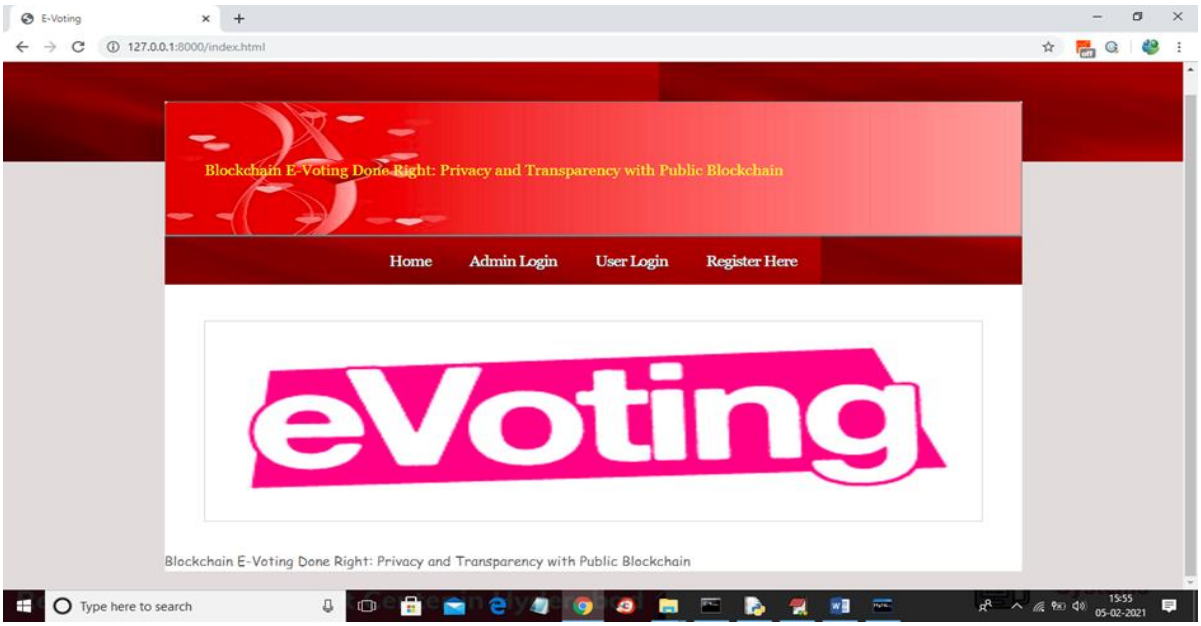
```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'EVoting.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

test.py

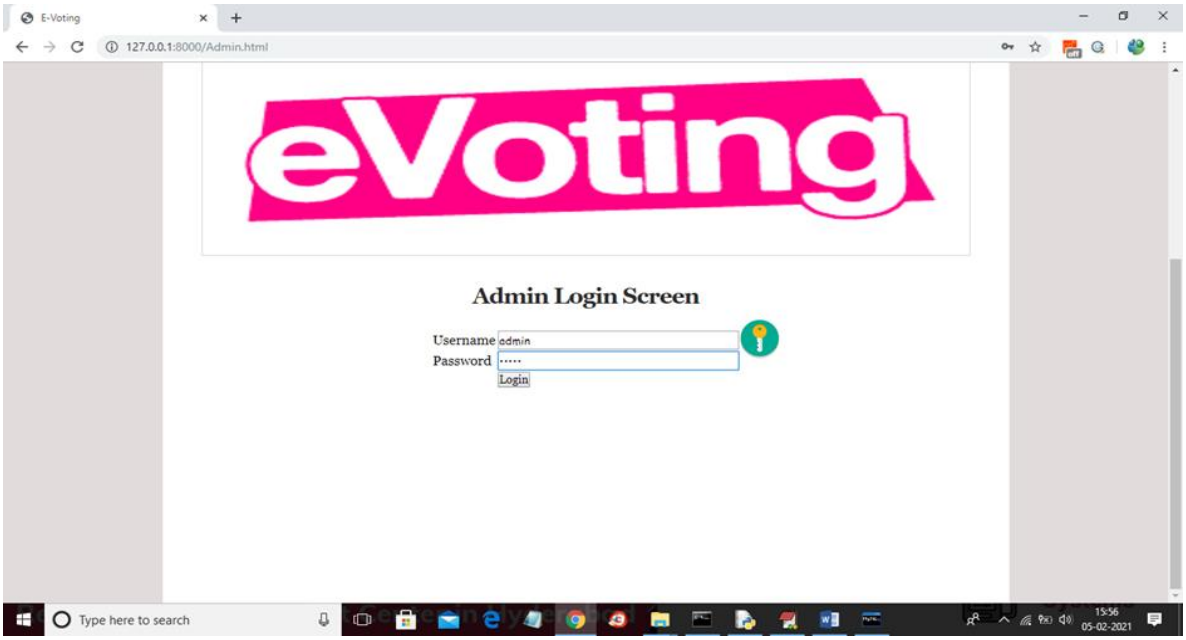
```
import os
import cv2
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
with open('model/model.json', "r") as json_file:
    loaded_model_json = json_file.read()
    classifier = model_from_json(loaded_model_json)
classifier.load_weights("model/model_weights.h5")
classifier._make_predict_function()
```

```
print(classifier.summary())

names = ['Azizullah Karimi', 'Hasibullah Atayi', 'khushhal Qasimyar', 'Nazif Mal',

    'pacha khan khogyani', 'Ramin paikar', 'SayedAhmad Seyar sawiz', 'Venket Rao', 'wahidullah Bdri', 'yahya Maqsidi']

face_detection = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

img = cv2.imread('test.png')

faces = face_detection.detectMultiScale(img,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags= cv2.CASCADE_SCALE_IMAGE)

print("==================="+str(len(faces)))

if len(faces) > 0:

   faces = sorted(faces, reverse=True,key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]

   (fX, fY, fW, fH) = faces

   roi = img[fY:fY + fH, fX:fX + fW]

   img = cv2.resize(roi, (64,64))

   im2arr = np.array(img)

   im2arr = im2arr.reshape(1,64,64,3)

   img = np.asarray(im2arr)

   img = img.astype('float32')

   img = img/255

   preds = classifier.predict(img)

   predict = np.argmax(preds)

   name = names[predict]

   name = name.lower()

   print(name)

   cv2.imshow('aa',roi)

   cv2.waitKey(0)
```

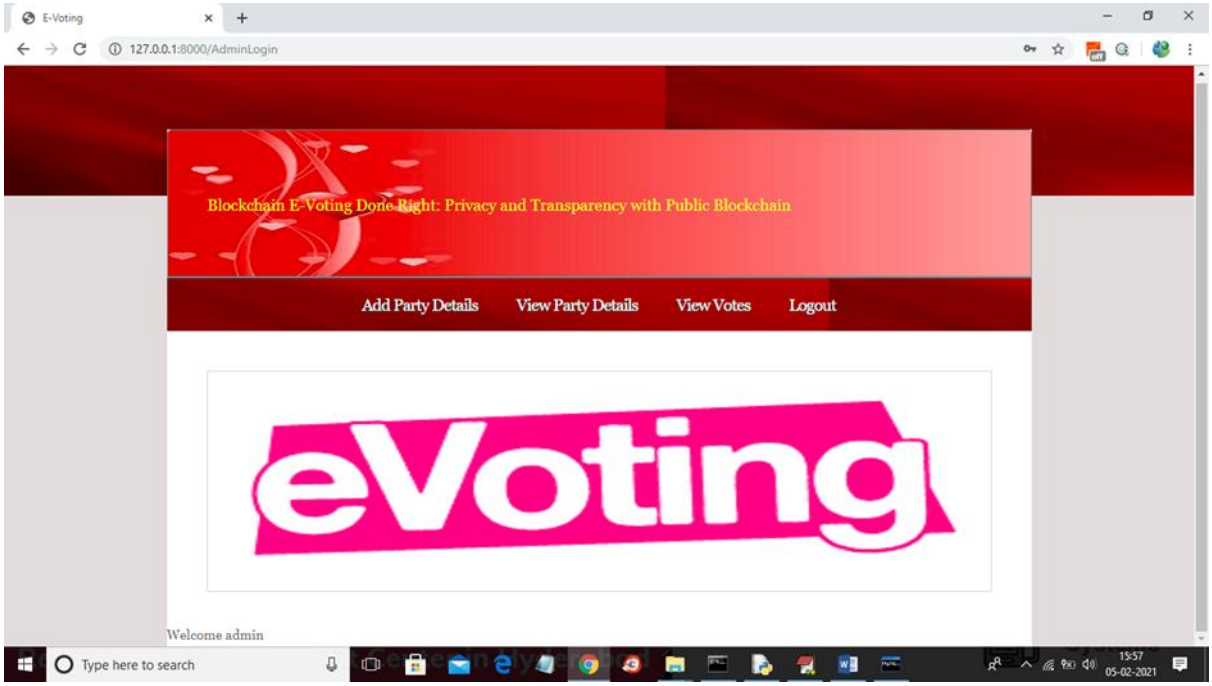
5. SCREENSHOTS



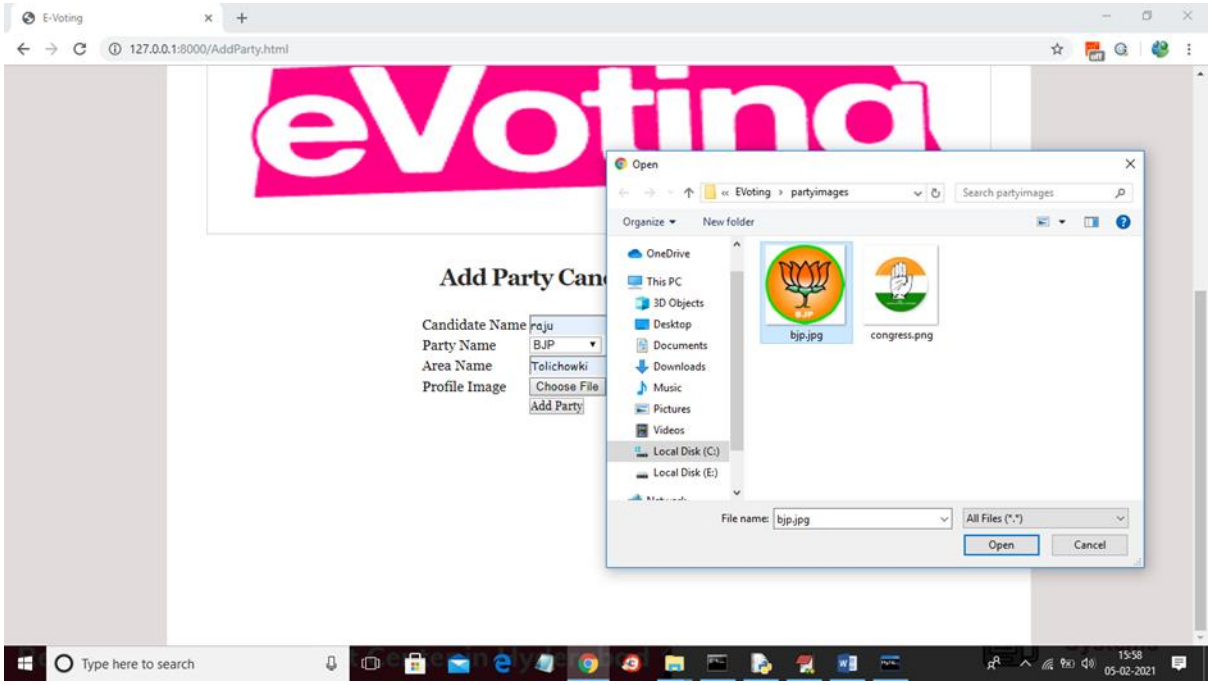
Screenshot 5.1: Click on 'Admin Login'



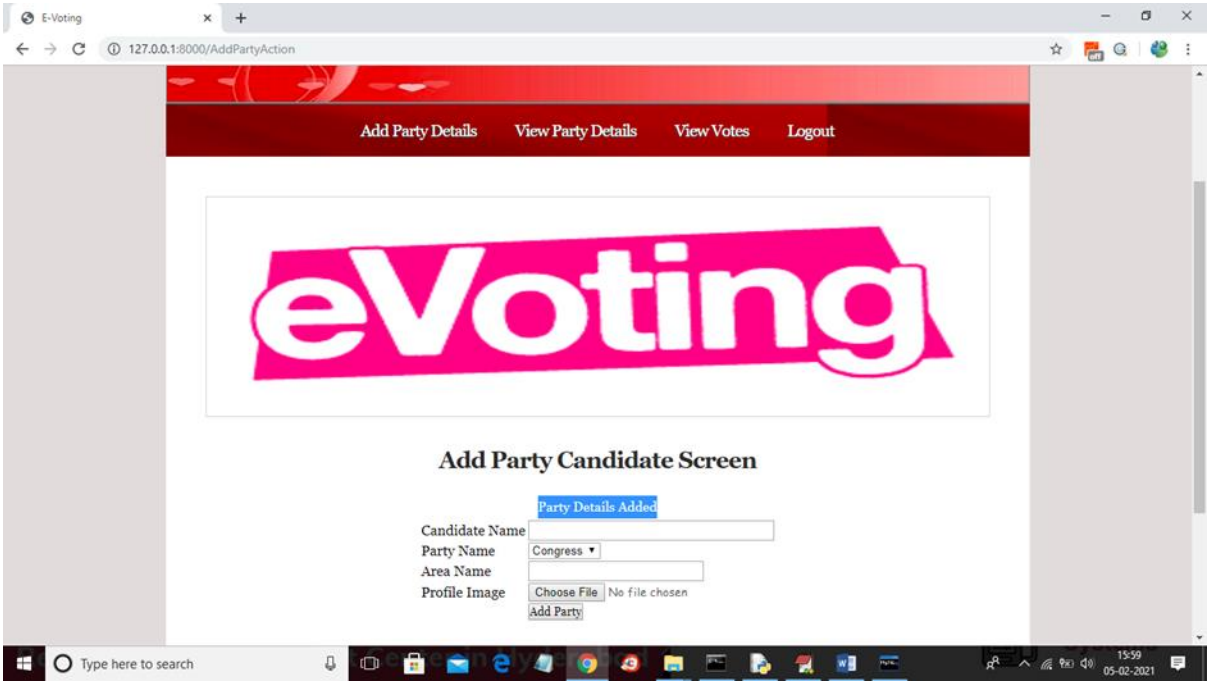
Screenshot 5.2: Enter 'Admin login' credentials



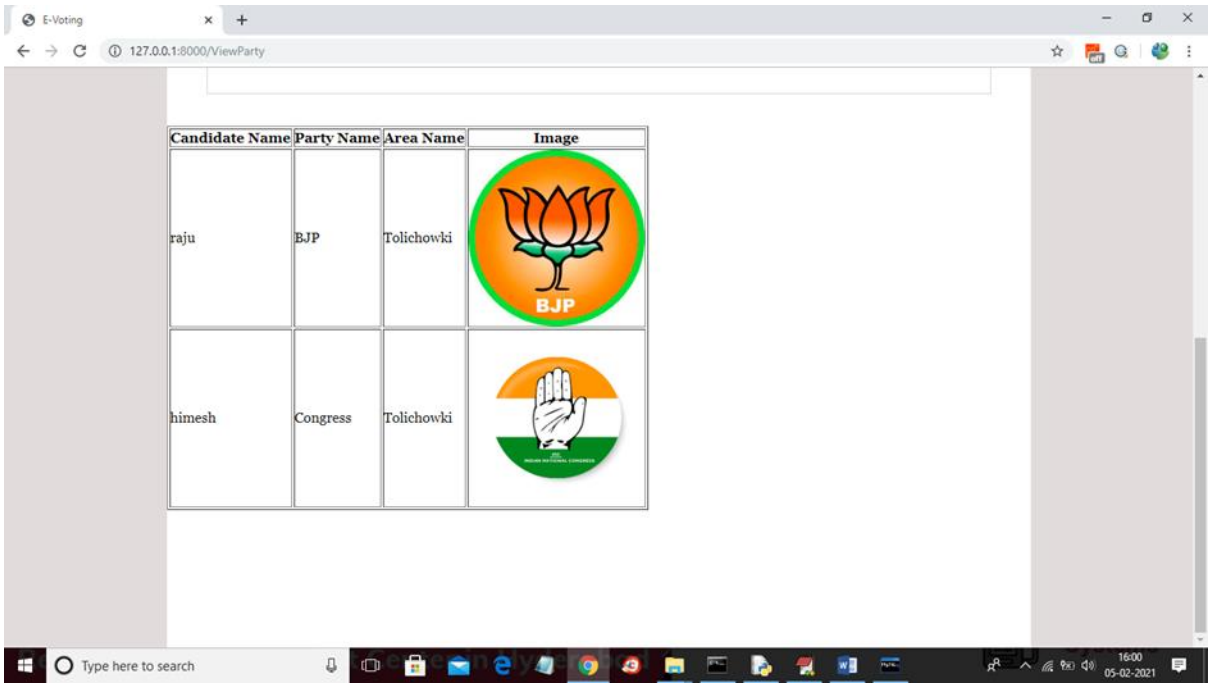
Screenshot 5.3: Successful 'Admin login' page



Screenshot 5.4: Add Party Candidate name

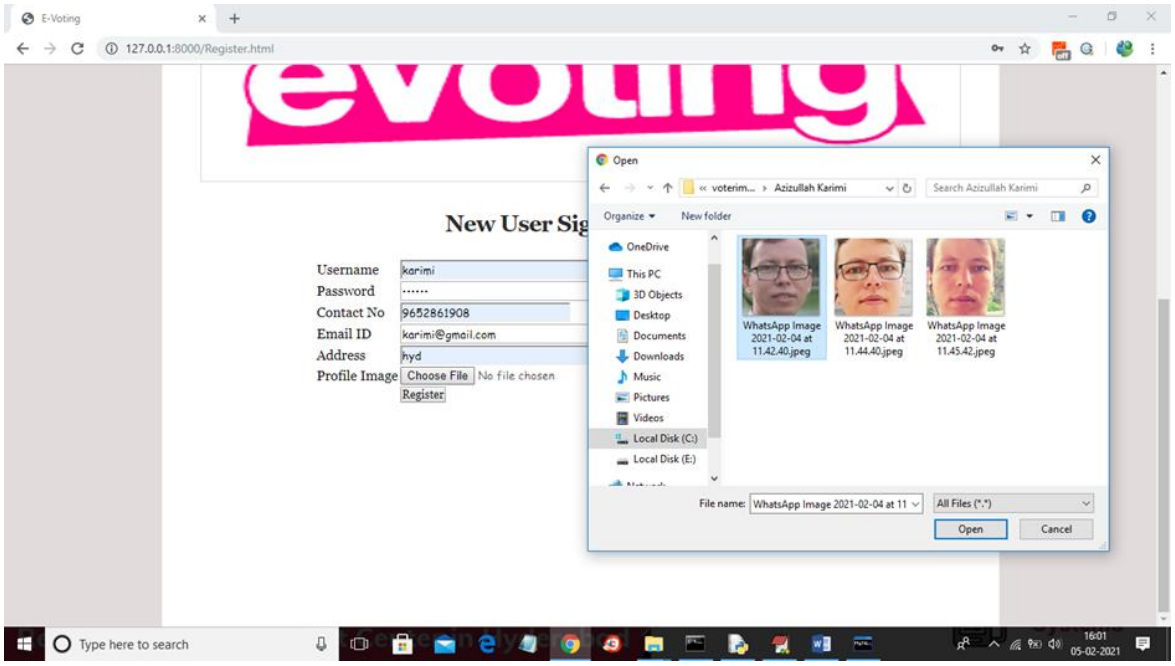


Screenshot 5.5: Party Candidate screen

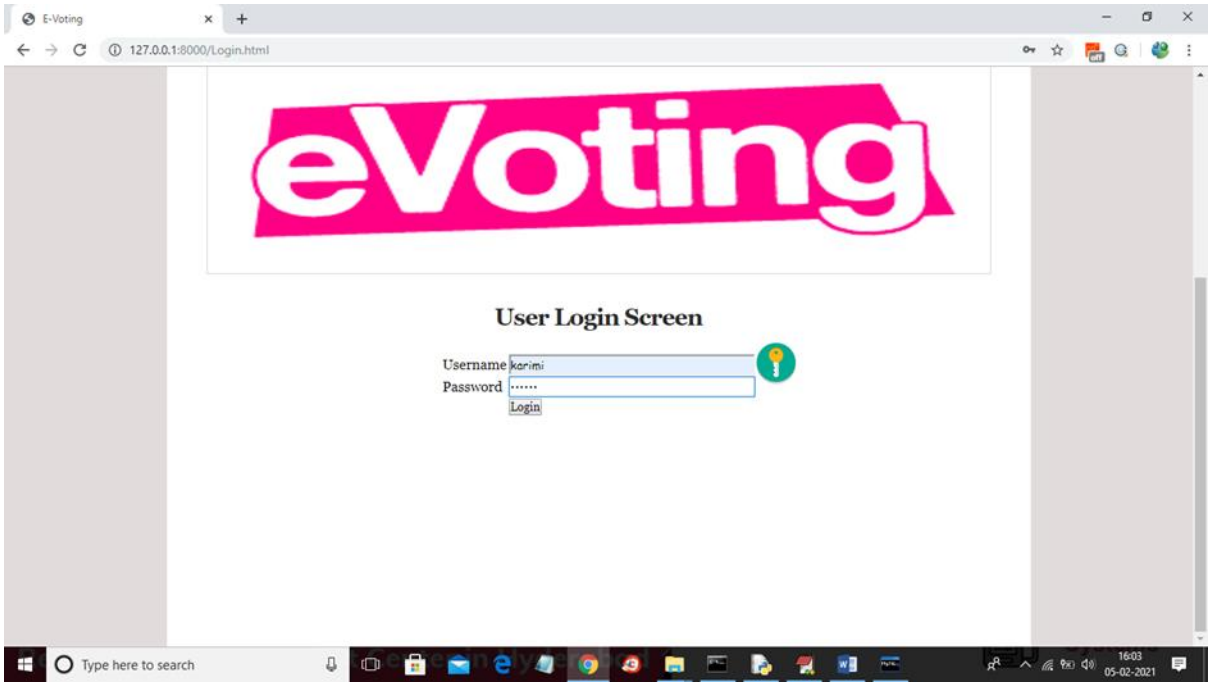


Screenshot 5.6: Registered Candidate Screen

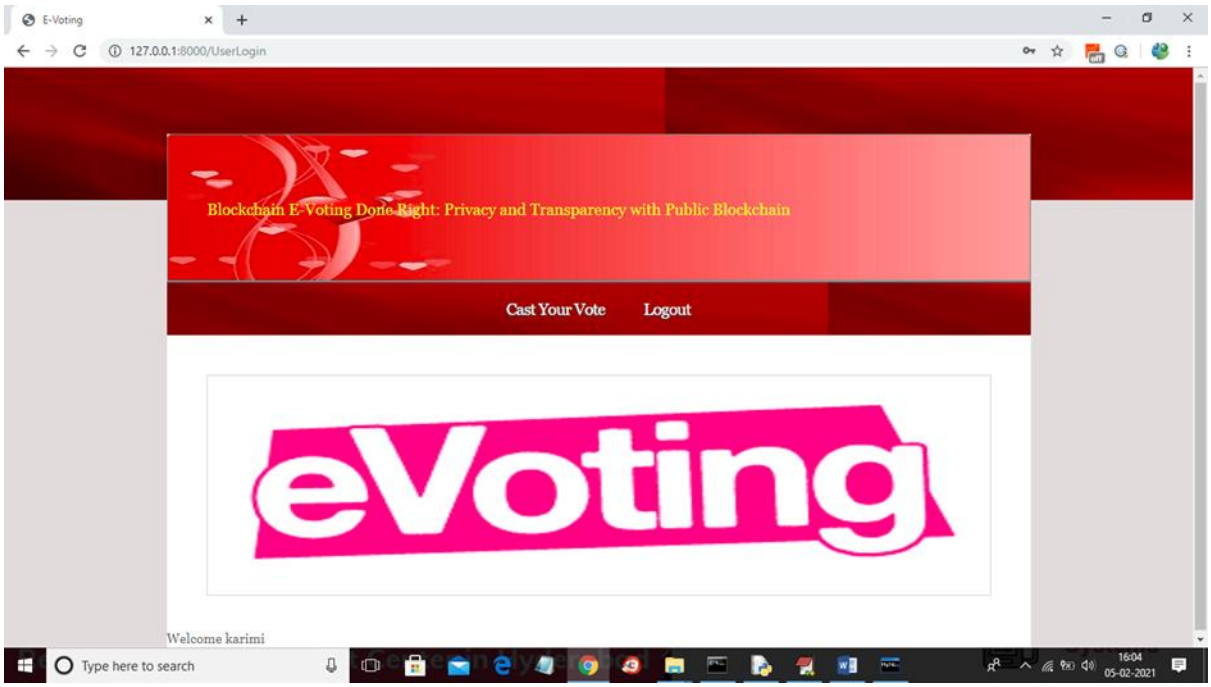
Screenshot 5.7: New User sign-up page for casting vote



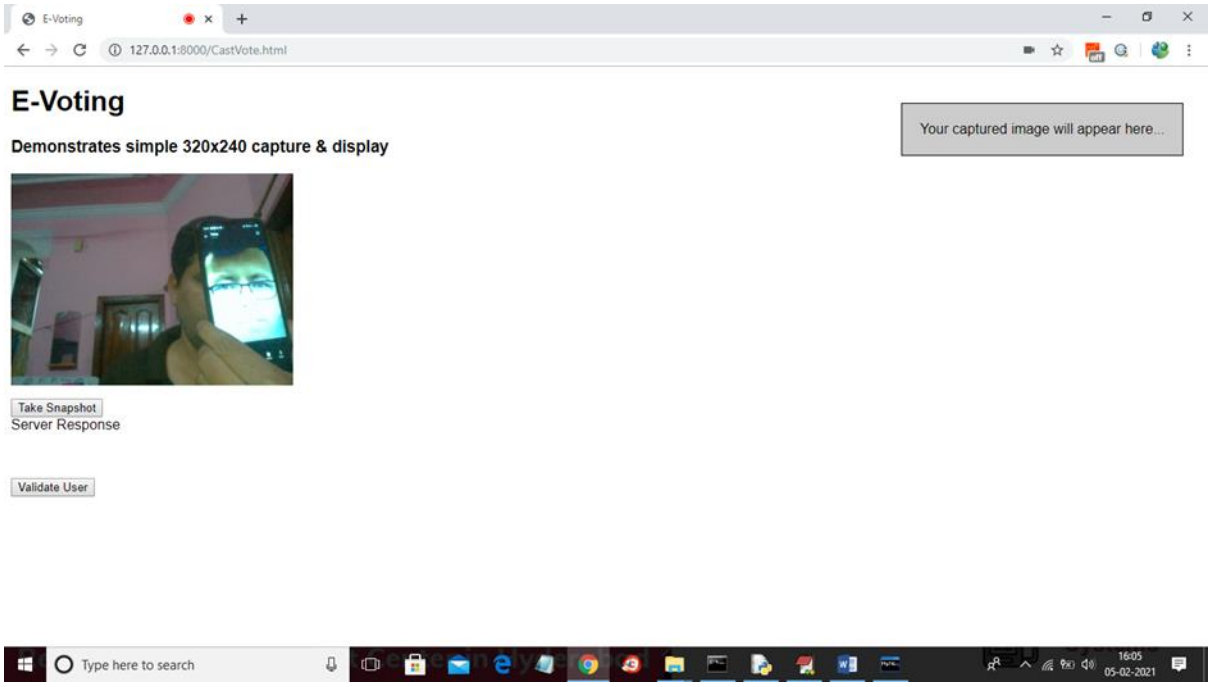
Screenshot 5.8: Entering registered user sign-up details



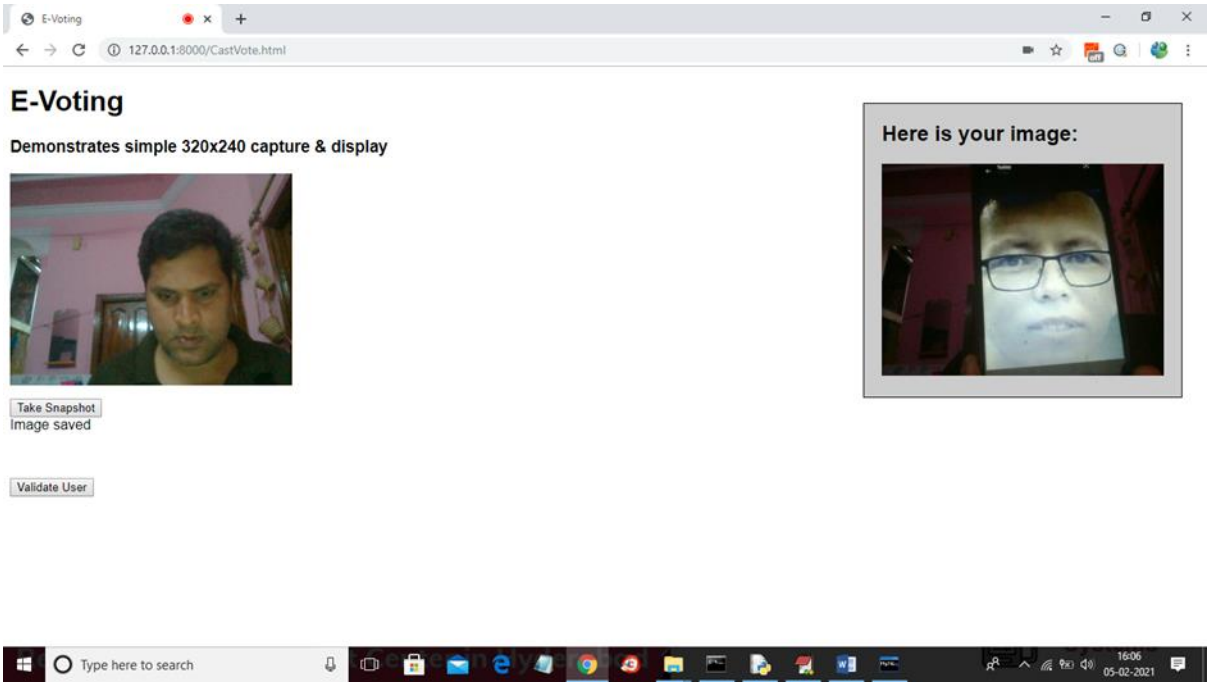
Screenshot 5.9: User login screen



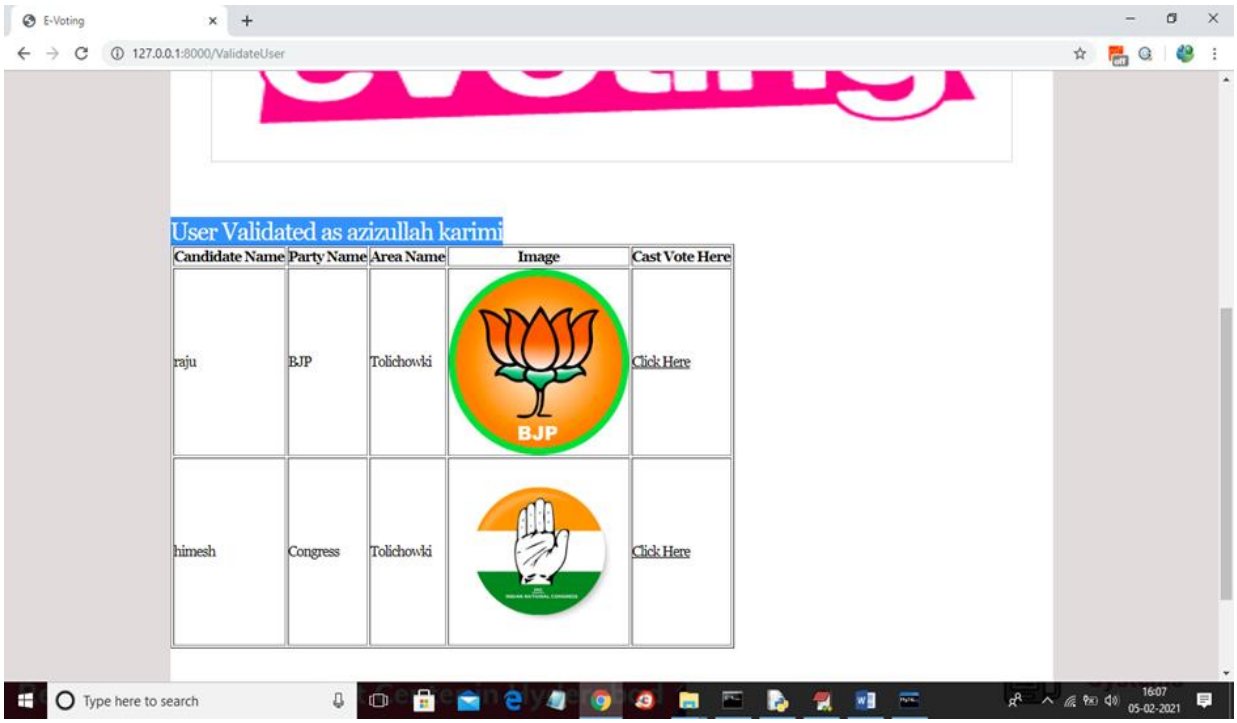
Screenshot 5.10: Vote casting page of registered user



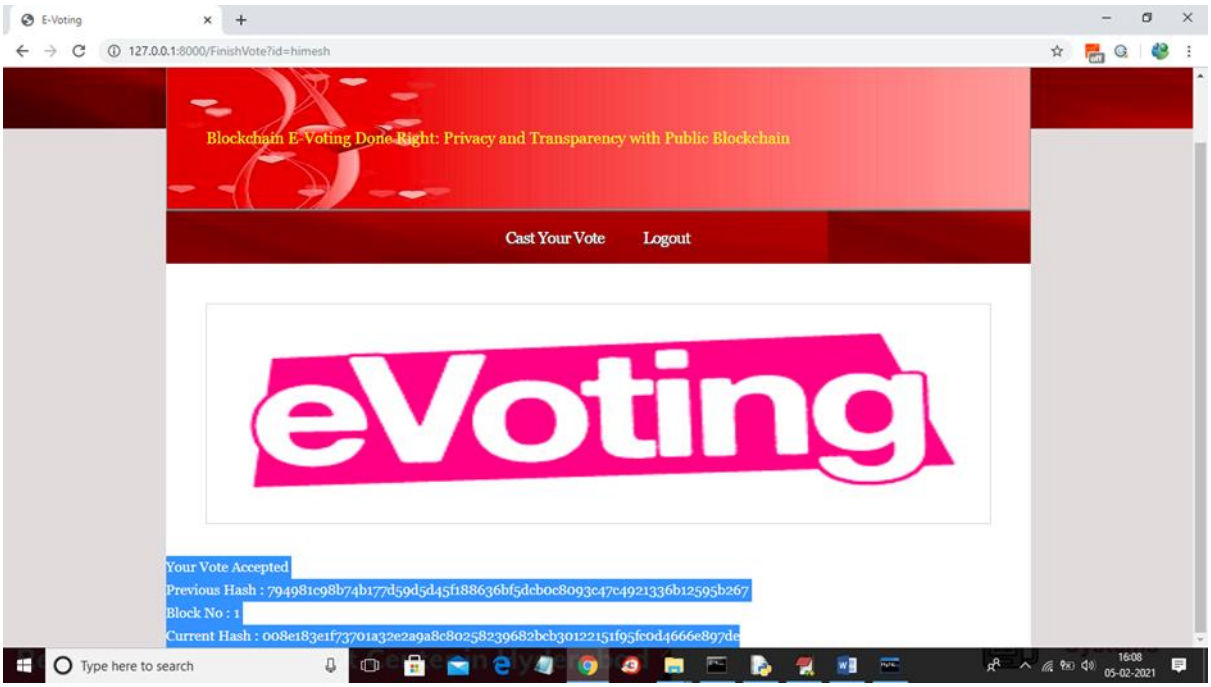
Screenshot 5.11: Face recognition of the Registered voter



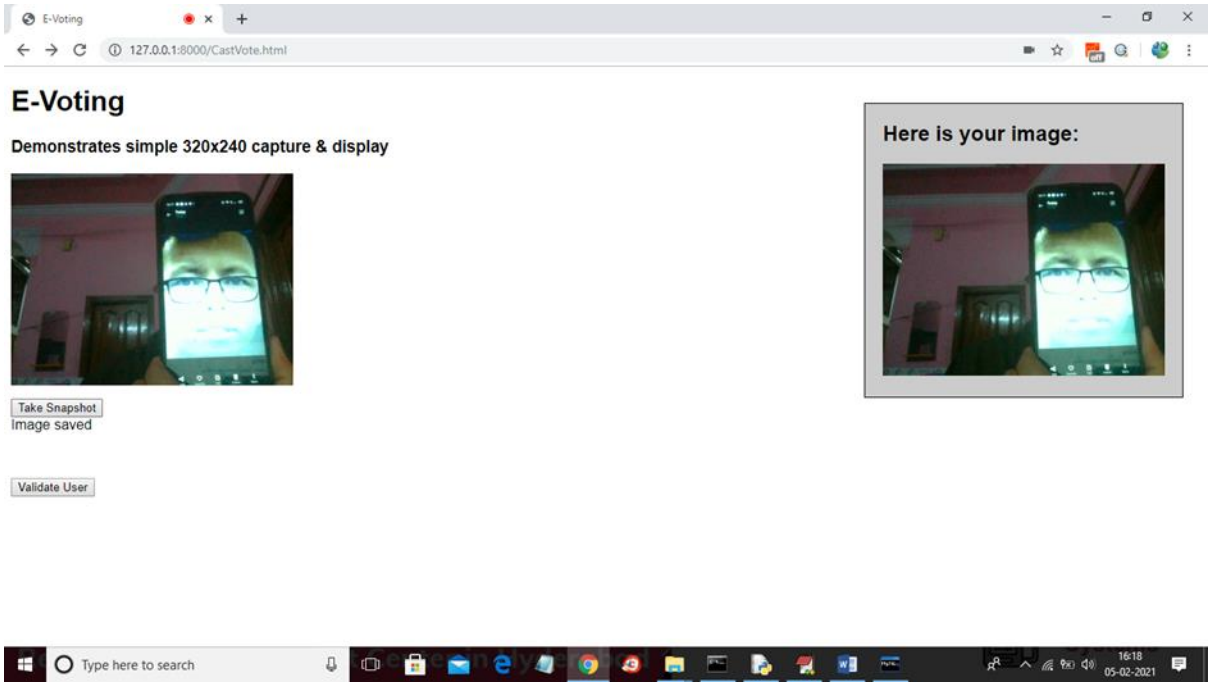
Screenshot 5.12: Capturing live image of the user



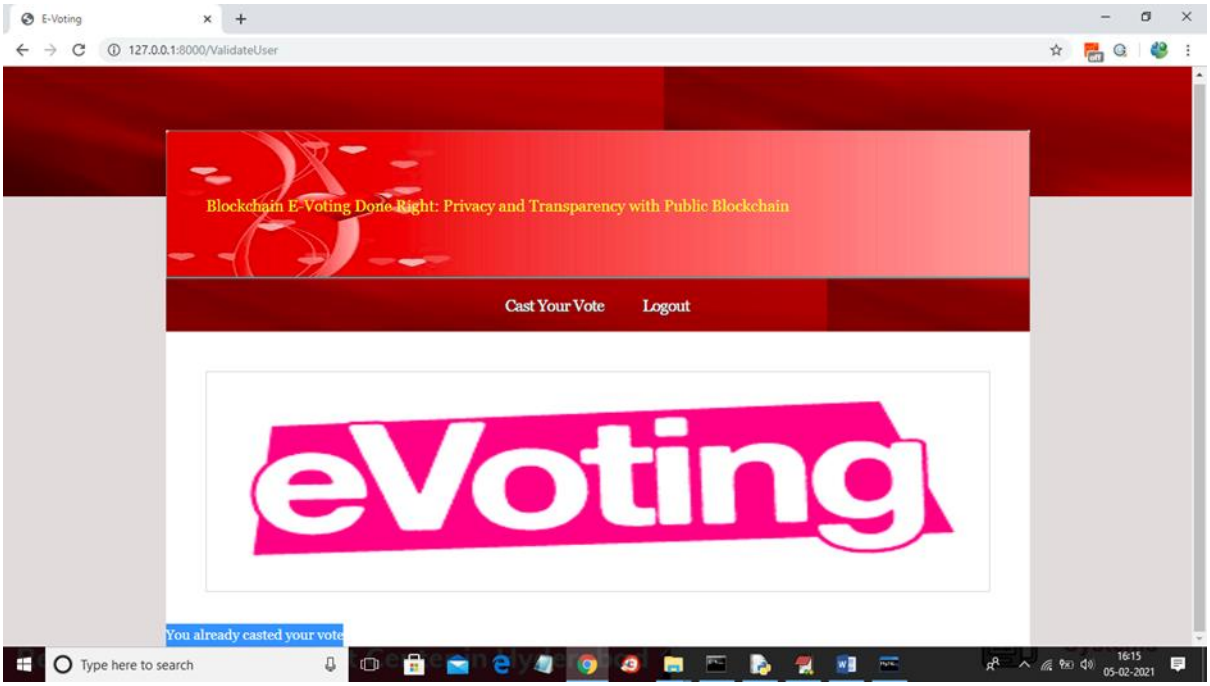
Screenshot 5.13: Voter casting vote for the desired party



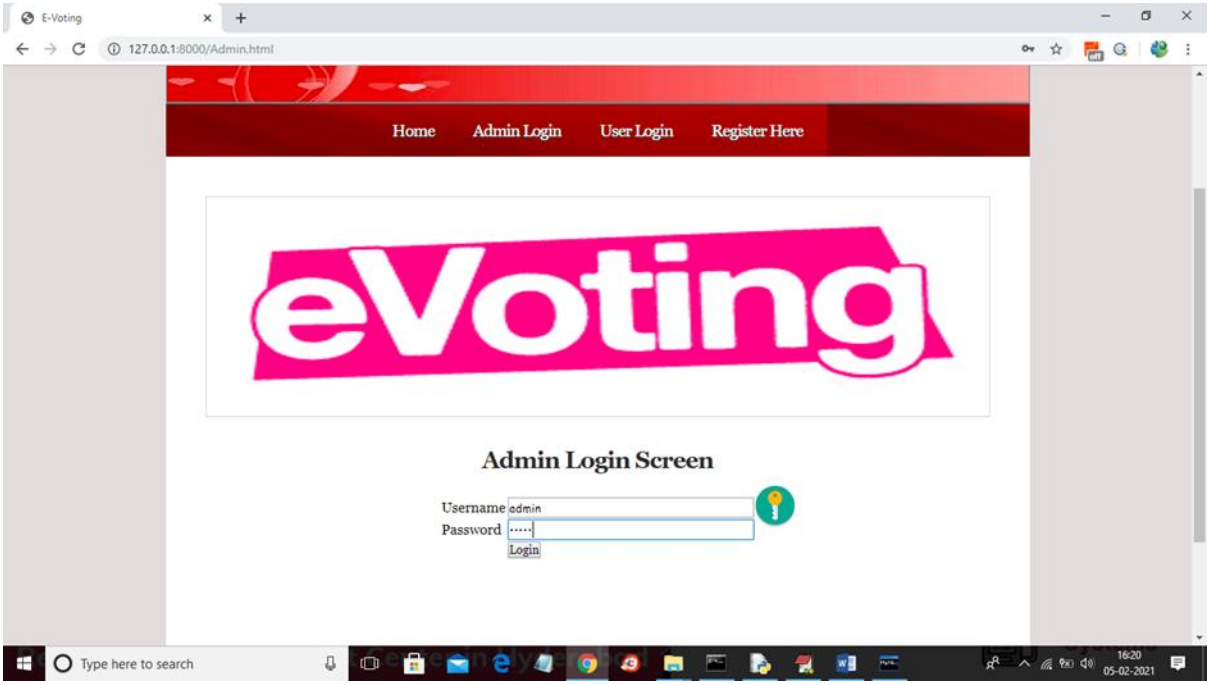
Screenshot 5.14: Successful casting of vote



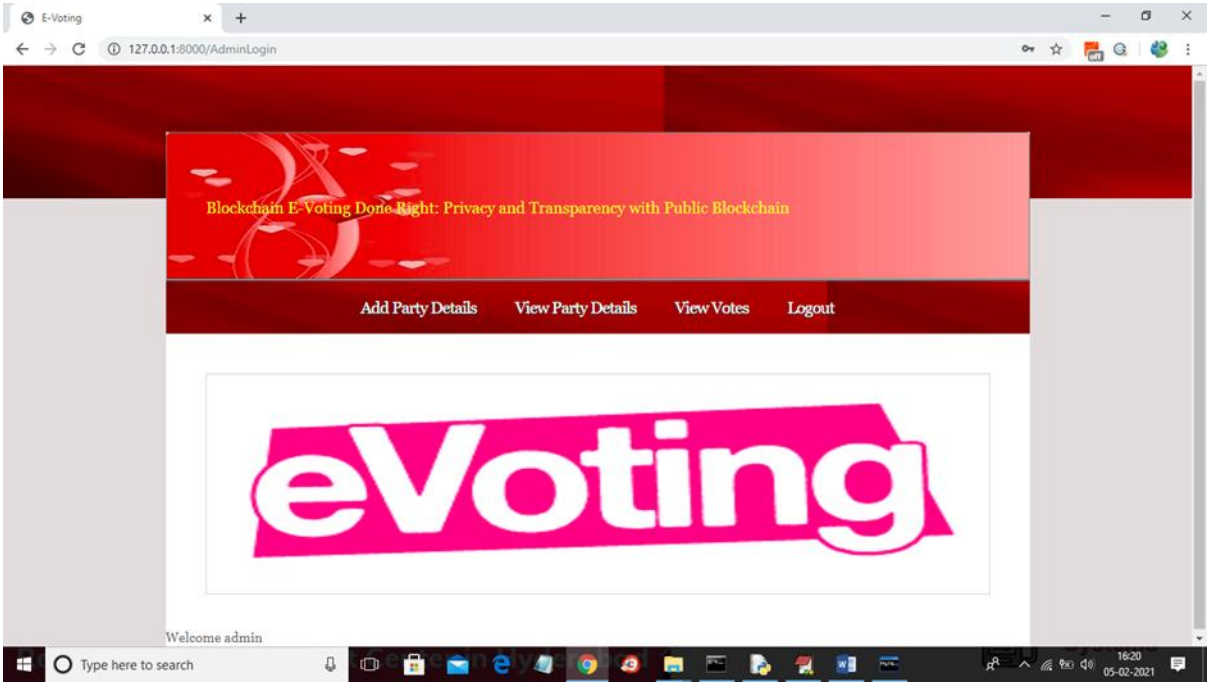
Screenshot 5.15: Same registered voter trying to vote again



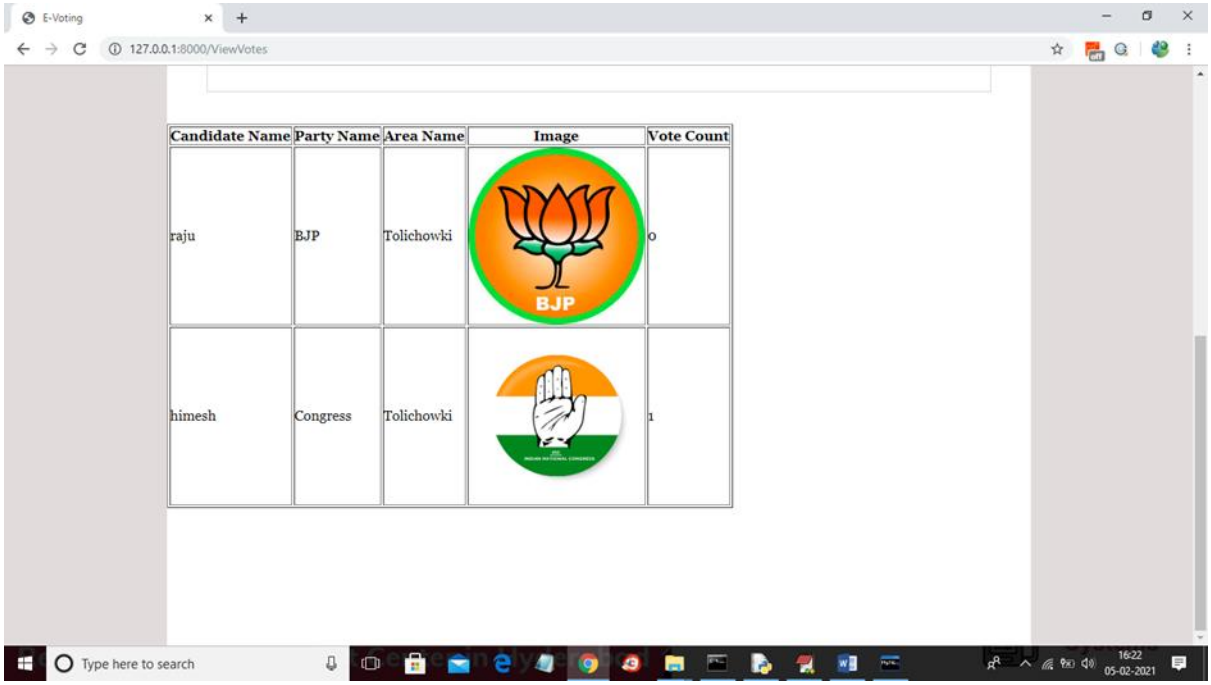
Screenshot 5.16: Page showing 'You already casted your vote' and user can logout



Screenshot 5.17: Log in to admin page to view the vote count



Screenshot 5.18: Admin welcome page to view the vote count by clicking on ‘view votes’



Screenshot 5.18: view count page after admin login

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of trying out is to discover errors. testing is the manner of trying to find out every achievable fault or weak spot in a work product. It offers a way to test the capability of components, sub-assemblies, assemblies and/or a finished product it is the system of exercising software with the rationale of ensuring that the software program system meets its requirements and consumer expectancies and does now not fail in an unacceptable way. there are numerous types of tests. every check type addresses a selected testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit trying out entails the layout of test instances that validate that the inner application logic is functioning nicely, and that software inputs produce valid outputs. All selection branches and inner code glide need to be verified. it is the trying out of man or woman software gadgets of the utility .it is carried out after the completion of an man or woman unit earlier than integration. that is a structural trying out, that is predicated on information of its creation and is invasive. Unit exams perform fundamental exams at aspect level and take a look at a selected enterprise method, utility, and/or gadget configuration. Unit tests ensure that every particular direction of a enterprise procedure plays as it should be to the documented specs and contains really described inputs and predicted outcomes.

6.2.2 INTEGRATION TESTING

Integration tests are designed to check included software additives to determine in the event that they surely run as one application. testing is occasion pushed and is extra involved with the fundamental final results of screens or fields. Integration tests show that even though the additives have been in my opinion delight, as proven by using successfully unit checking out, the aggregate of components is correct and constant. Integration checking out is mainly geared toward exposing the troubles that get up from the mixture of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is cantered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Admin Login	To Access the Admin page	Enter admin login credentials	Valid Login credentials allows the access and invalid credentials denies access
2	User Login	To Access the User page	Enter user login credentials	Valid Login credentials allows the access and invalid credentials denies access

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Although we can see slight differences in network times, they are so negligible that public blockchain has more advantages in such an electoral system due to its openness of data and that anyone can watch them in the real time. A private blockchain is a bit faster, but it reduces the credibility of the whole system by being partially centralized because it only runs where the authority wants it. The table shows that the average times to add one person's voice are: Ganache 6.32 s (median 6.34 s), Hyperledger Composer 6.05 s (median 6.04 s), and Ethereum Ropsten 17.75 s (median 17.93 s). These times are influenced by the used consensus algorithm and also by the block time

7.2 FUTURE SCOPE

Future Enhancement is being planned to further analyze and enhance the protocol to a private blockchain is a bit faster, but it reduces the credibility of the whole system by being partially centralized because it only runs where the authority wants it.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] N. K shetri and J. Voas, “Blockchain-Enabled E-Voting,” IEEE Software, vol. 35, pp. 95-99, July 2018.

[2] M. Pawlak, J. Guziur, and A. Poniszewska - Maranda, “Voting Process with Blockchain Technology: Auditable Blockchain Voting System,” in Lecture Notes on Data Engineering and Communications Technologies, vol. pp. 233-244, Springer, Cham, 2019.

[3] B. Singhal, G. Dhameja, and P. S. Panda, “How Blockchain Works,” in Beginning Blockchain, pp. 31-148, Berkeley, CA: Apress, 2018.

[4] Agora, “Agora Whitepaper,” 2018.

[5] R. Perper, “ Sierra Leone is the first country to use blockchain during an election - Business Insider,” 2018.

[6] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” tech.rep., 2008.

[7] G. Wood et al., “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum project yellow paper, vol. 151, pp. 1-32, 2014.

[8] S. Landers, “Netvote: A Decentralized Voting Platform - Netvote ProjectMedium,” 2018.

[9] P. McCorry, S. F. Shahandashti, and F. Hao, “A Smart Contract for Boardroom Voting with Maximum Voter Privacy,” in Lecture Notes in Computer Science, ch. FCDS, pp. 357-375, Springer, Cham, 2017.

[10] Z. Brakerski and V. Vaikuntanathan, “Efficient Fully Homomorphic Encryption from (Standard)LWE,” SIAM Journal on Computing, vol. 43, pp. 831-871, January 2014.

8.2 GITHUB LINK

https://github.com/Ar27-Rehman/E-voting/tree/main/EVoting/EVoting