A Project Report

on

DECENTRALIZED VOTING SYSTEM

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

B.Tech(CSE)

Under The Supervision of
Dr.Anup Kumar Sharma Professor

Submitted By
Shweta-19SCSE1010684
Rishab Sharma -19SCSE1010186

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGGALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA MAY,
2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"A web app for Decentralized Voting System"** in partial fulfillment of the requirements for the award of the BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of June-2022 to May 2023, under the supervision of Dr. Anupam Kumar Sharma Professor Department of Computer Science and Engineering, Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shweta 19SCSE1010684

Rishab Sharma 19SCSE1010186

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Anupam Kumar Sharma
Professor

CERTIFICATE

The ETE Project Viva-Voce examination of Shweta and Rishab Sharma has been held on_____and his work is recommended for the award of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

ABSTRACT

On-line vote casting in elections may seem like a logical leap forward thinking about the many other every day activities, like banking and shopping, that we complete on-line. However, voting on line does present precise challenges that generally don't observe to other internet- based methods. those challenges are associated with a spread of things, consisting of the security required for online balloting, prison requirements and frameworks, public opinion, and investment.

Elections always require a excessive level of safety in an effort to defend voter privateness and the integrity of final results. This studies is aimed to layout a decentralized e-balloting gadget. The core idea is to combine the blockchain generation with secret sharing scheme to be able to comprehend the decentralized e-vote casting software without a relied on third party. It offers a public and obvious voting process while shielding the anonymity of voter's identification, the privateness of facts transmission and verifiability of ballots throughout the billing section.Meeting the safety desires of elections means that on line balloting generation should overcome boundaries that don't apply to different on line-based procedures, so right here we give you the idea of "decentralized voting system" a good way to be made with the assist of solidity for writing smart contracts, flutter that we'll be using for frontend and node.js with a purpose to be use for backend.As a end result we'll have an app for digital voting structures in which our records might be saved in friends and backend can't be altered by using a unmarried entity as a count of security and just like every other thing we can cast our vote on-line and that too with proper validation and no threat of tampering.

List of Tables

Table for Student Data:

S.No.	Name	Enrollment Number	Admission Number	Program/ Branch	Sem
1.	Shweta	19021011833	19SCSE1010684	B.Tech/CSE	8
2.	Rishab Sharma	19021011379	19SCSE1010186	B.Tech/CSE	8

Table for Faculty Data:

Guide Details:-

S.No.	Name	Designation	Contact
1.	Dr. Anupam Kumar Sharma	Professor	9319394901

Table of Contents

CHAPTER -1
Introduction

Voting is a method for an electorate to express their opinions or make a decision collectively. Election campaigns, debates, and discussions are typically followed by voting. While voting, the person standing in elections to get selected is known as a candidate and voter refers to the individual casting a ballot for that candidate. Typically, a voter has the option to select candidates from the list or anyone else they choose. To prevent others from learning who a citizen is voting for, voting ballots must be unsigned and marked by the electorate in private cubicles[2].From the 17th century, voting has been the conventional means by which contemporary consultative democracy has operated.Voting is also utilized in a wide range of private organizations and businesses, including golf equipment, associations, and organizations.Voting, mail, and other traditional offline services are moving online due to the internet's and information technologies' rapid advancement. Digital voting is the term used to describe casting votes online (e-vote casting). Voting is done electronically, as is the counting of votes.

The electorate and electoral authorities use electronic voting. The voter may electronically cast their ballots and do so from any location and submit them to the election officials. Voter registration is the responsibility of the electoral government. E-voting, which is increasingly preferred over traditional voting because of its great

efficiency and versatility, can save both time and effort. E-voting became a crucial tool for many businesses as the internet developed.

1.1 Importance of e-voting system:

In all industries, the epidemic has spurred digitization. The experience has occasionally been traumatic. In contrast, it has been a true revelation in some cases, leading people to ask why we hadn't done things this way previously. Without a doubt, this applies to voting online[13].

Every year, or more frequently, organizations of all kinds need to come to agreements that are enforceable in court. organizations of all sizes, co-ops, associations, sports teams, or local communities. everything from union elections to the approval of professional organizations' financial statements. In the past, decisions were made through time-consuming in-person meetings, best case scenario with the option of voting by mail.

It makes complete sense that voting must move to a digital platform as our lives move more and more online.Although public elections have not yet adopted such techniques, some commercial groups have quickly embraced internet voting, allowing them to benefit from important advantages[14].Thanks to the availability of online voting on a variety of platforms, including computers, tablets, and smartphones, the full voting procedure can be accomplished by pushing a button or tapping a screen.This suggests that everyone on the planet is eligible to vote. The days of wasting a whole meeting trying to figure out when everyone will be in town for the election are long gone.Members might, for example, use

the software while traveling and cast their votes while lounging on the beach. Larger clubs that have members all around the country also greatly benefit from this.It eliminates the hassle of having to find and reserve numerous polling places or shell out a lot of money for numerous mail-in ballots. During voting hours, every member may vote from the comfort of their own home[17].

Most reliable election service companies build cutting-edge security into their software.From nomination to tabulation, the entire process is protected by a secure network. Common features include ballot tracking, which records the precise moment a ballot is processed, secret ballots, which keep ballots private, single-vote verification, which makes sure that members don't unintentionally cast more than one vote, and secret ballots, which prevent ballots from being disclosed. Furthermore, some service providers guarantee that they will never share member information with any outside companies[15].

Very likely, accuracy ranks with security as the top perk of online voting. The outcome of a vote can be tipped by a slight counting inaccuracy. Security measures for online voting ensure that every vote is swiftly counted. Vote tracking records the number of times a vote is opened or clicked, as well as the precise minute it is carried out. Every action is documented, making it possible to get precise information right away. As the final tabulation process is completely auditable, the accuracy may be verified twice in case there are any questions.

More members are more likely to vote as a result of improved accessibility and usability, especially if they can do it on their smartphone from the comfort of their couch. The ability

to integrate social media into a number of online voting software choices will help your organization increase participation even further. A page will appear asking people to share their findings on Facebook or LinkedIn after the voting is complete. In the newly formed post, only the fact that they cast a ballot will be made public, not who they voted for. This might encourage more people to take part[16].

1.2 Formulation of problem

A number of security threats affect voting systems now in use, such as electronic voting or ballot boxes[7], including DDoS attacks, poll booth theft, vote tampering and manipulation, virus assaults, etc. Also, they require a lot of paperwork, staff time, and other resources. As a result, existing systems develop a mistrust.[5].

Some of the drawbacks include:

- Election day long lines.
- Security breaches such as vote rigging and data leakage.
- It is less efficient and takes more time because there

  is a lot of documentation needed.
- It is challenging for voters with disabilities to get to the polls.
- Election expenses are very expensive.

1.3 SOLUTION:-

The security, dependability, immutability, and transparency of voting systems can all be enhanced by blockchain technology.

Suppose you are a registered voter who casts your ballot using an electronic voting machine (Electronic Voting Machine)[6]. You might not be able to tell whether your vote went to the candidate for whom you cast it or whether it was misdirected into the account of another candidate because it is, after all, a circuitry, and if someone tampers with the microchip.because it is impossible to determine how you voted[4]. However, if you use blockchain, it records everything as a transaction, which will be detailed in more detail below. As a result, you receive a receipt for your vote (in the form of a transaction ID), which you can use to verify that it was securely counted.

In our proposed system project named Decentralized Voting System App is built on Solidity's features. It puts into action a voting contract. The biggest challenge with electronic voting is preventing the assignment of duplicate votes, thus we'll be using Node JS for backend programming to address this.

Advantages of this project:-

- You can vote anywhere / anytime (during pandemics like- covid 19 where it is impossible to hold elections.
- Secure
- Faster

1.3.1 Tools and Technologies Used:

- **Smart contacts** for solidity
- **Flutter** for frontend
- **Node.js** for handling backend of the app

CHAPTER-2
Literature Survey

The most significant feature of a democracy is that it gives individuals the opportunity to express their unique voices by participating in the decision-making process rather than just exchanging ideas, opinions, and goals. Yet, for voting to function as intended, there must be a clear, secure method that enables participants to consciously maintain their anonymity [3]. Creating a system that prevents improper use of the amassed data and achieves preferred openness in the security measures implemented to protect voter privacy, the accumulated effects, and eventually democracy itself is the challenge[1].

E-voting is a quick and inexpensive way to conduct a voting operation. Benefits of it include real-time, data-rich, and high security requirements. Yet, concerns over communication privacy and Internet security have grown more serious. Anonymity is necessary for electronic voting, which encryption cannot offer on its own. A vote shouldn't be connected to the voter in electronic voting, for example[20]. In order to complete the voting process, electronic votinginvolves computers, mobile devices, and the internet. This branch of cryptography research makes use of the fundamental encryption and signature techniques. In the disciplines of business and information security, the creation of a more practical and safe electronic voting system is a prominent topic[18].We propose approaches for integrating blockchain in future e-voting systems to enhance the security and anonymity of electronic voting.E-voting systems(EVS)have capacity advantages over many of the current voting procedures.

The main circumstances in these structures are protection, transparency, accuracy and dependability[12]. EVS continues to expand since the technology is affordable and environmentally friendly because the resources are reused. This suggested device allows for the quick and accurate computation of results while maintaining voter privacy. To achieve security and reliability, we use a secret sharing mechanism and comfortable multi-celebration computation (SMC).

A digitalized democratic voting system could be the next revolutionary step towards a transparent and reliable election system, given how quickly blockchain technology is being adopted[10]. The potential for a decentralized voting system is discussed in this thesis along with any potential privacy, accuracy, and integrity problems. Using various smart contracts and the Ethereum blockchain, a proof of concept prototype was produced. The thesis also includes research on the prototype's social and environmental consequences as well as studies on electronic voting in general.Throughout the course of the project, it was evident that the suggested design would not work in practice and entails significant tradeoffs[8].

The report makes clear the issues that must be resolved before a blockchain-based voting system can be put into place.

2.1 Project Design

2.1.1 Proposed Approach

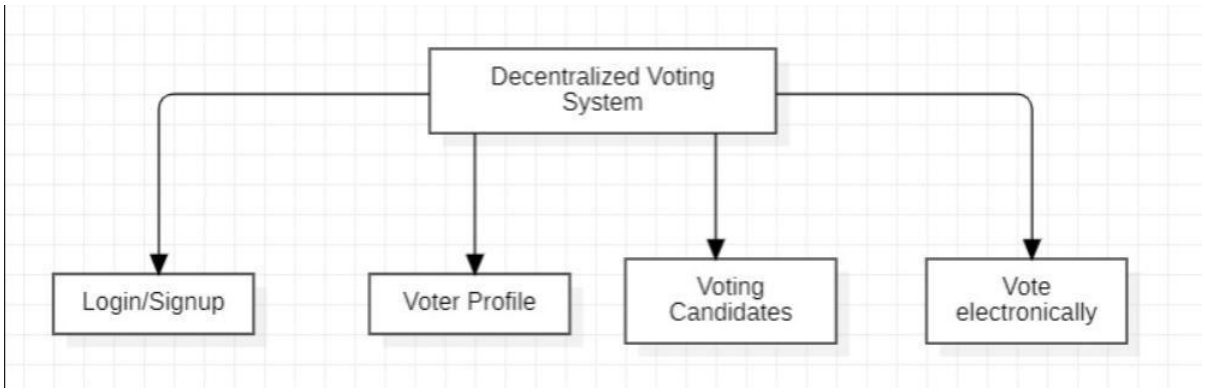


Fig.2.1 Proposed Diagram

2.1.2 Flowchart

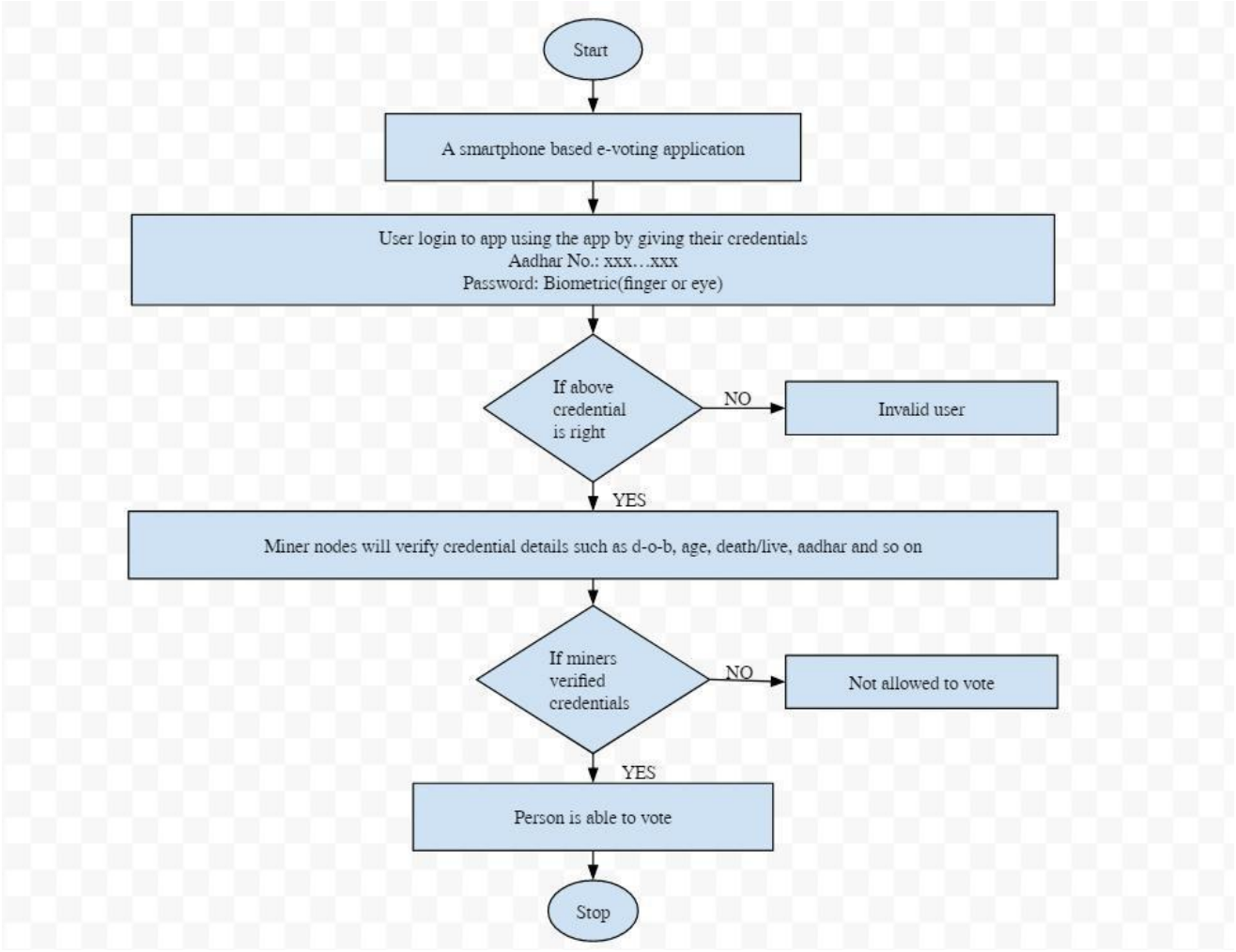


Fig.2.2 Flowchart

2.1.3 DATA FLOW

DIAGRAM- Level 0-

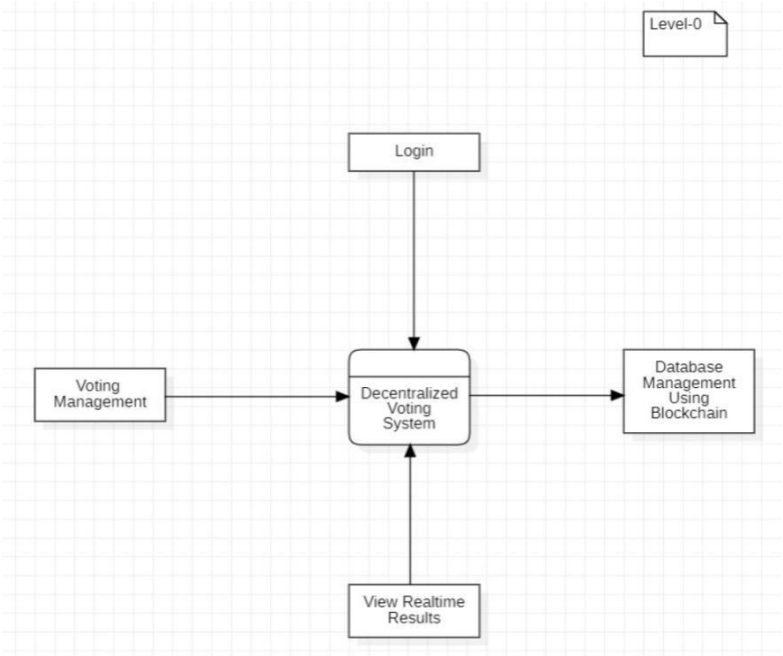


Fig.2.3 DFD level-0

Level 1-

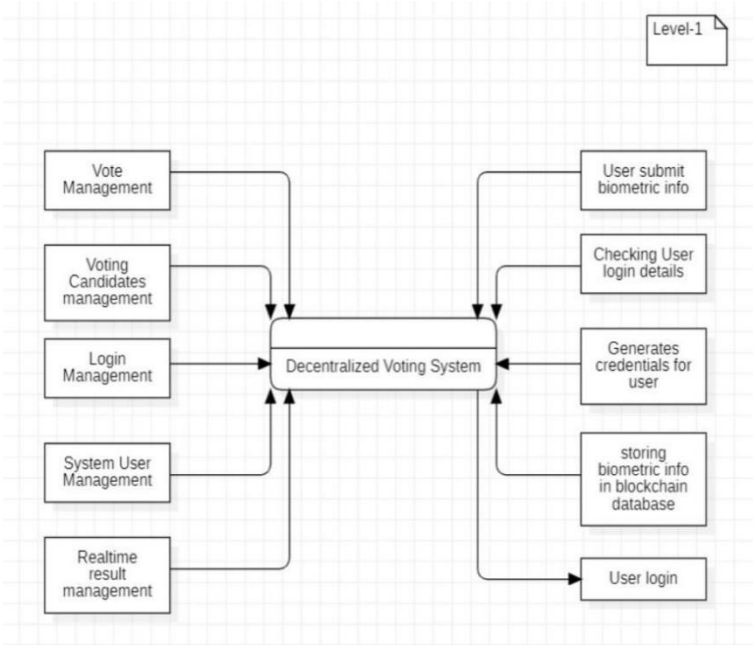


Fig.2.4 DFD level-1

Level 2-

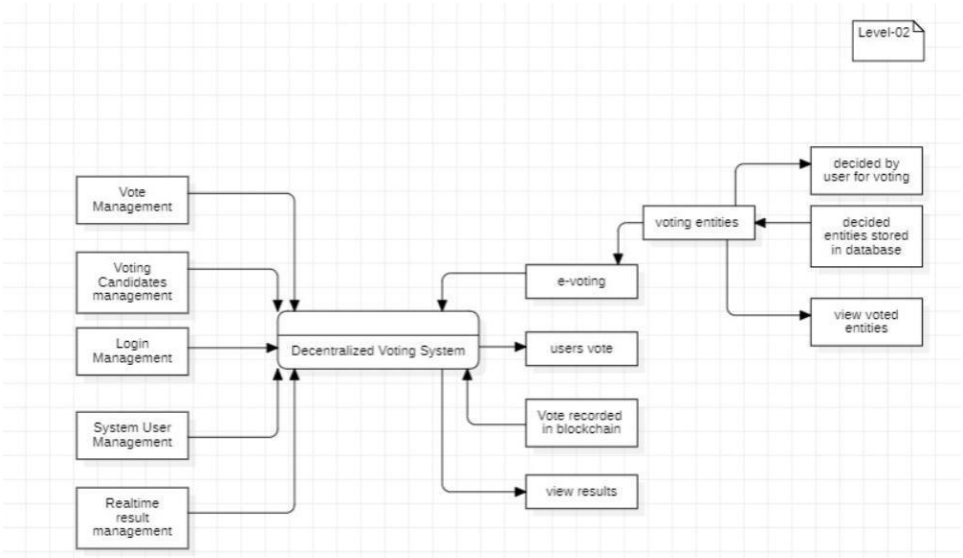


Fig.2.5 DFD level-2

CHAPTER-3
Methodology /Implementation

In blockchain computation done by the machines in the network. This computation is done to solve the complex problems/algorithms which are required to edit the blockchain. This is called peer-to-peer as all the machines are connected and each machine has the data. Here is our project for now we are using rinkabyte test network which has schema of these blockchain database and we dont actually need to do the computation. When the project goes live all the devices in the organization/authority will be used to solve the problems/algorithm. These complex problems will be generated as the network will grow the computing power from these peers/devices will be used to solve them and new data will be added to this blockchain network other blockchain networks such as etherium and bitcoin can be used but as these tokens has values each casting of vote will be expensive so we will have to use our own network. One alternative to is is using Polygon or solana network as these networks dont charge any thing but it will not be reliable for now, as they are still in beta state of their development and for sensitive thing like voting we need to have all control under the service provider.

Code for Creating Smart Contract:

```
pragma solidity 0.6.6;
```

```
contract Voting{


int alpha;

int beta;


constructor() public

{ alpha = 0;

beta = 0;

}


function getTotalVotesAlpha() view public returns(int) {

return alpha;

}


function getTotalVotesBeta() view public returns(int){

return beta;

}

function voteAlpha () public{
alpha = alpha+1;
}

function voteBeta () public{
beta = beta+1;
}
}
```

Solidity program to demonstrate:

```
pragma solidity 0.5.11;
// Smart Contract for the Voting application
contract VotingForTopper {

        // Refer to the owner
        address owner;

        // Declaring the public variable 'purpose'
        // to demonstrate the purpose of voting
        string public purpose;

        // Defining a structure with boolean
        // variables authorized and voted
        struct Voter{
                bool authorized;
                bool voted;
        }

        // Declaring the unsigned integer
        // variables totalVotes, and for the
        //3 teams- A,B, and C
        uint totalVotes;
        uint
        teamA;
        uint teamB;
        uint teamC;
```

```
// Creating a mapping for the total Votes
        mapping(address=>Voter) info;

        // Defining a constructor indicating
        // the purpose of voting
        constructor(
        string memory _name) public{
                purpose = _name;
                owner = msg.sender;
        }

        // Defining a modifier to
        // verify the ownership
        modifier ownerOn() {
                require(msg.sender==owner);
                _;
        }

        // Defining a function to verify
        // the person is voted or not
        function authorize(
        address _person) ownerOn public {
                info[_person].authorized= true;

        }

        // Defining a function to check and
        // skip the code if the person is already
```

```
// voted else allow to vote and
// calculate totalvotes for team A
function temaAF(address _address) public {
        require(
        !info[_address].voted,
        "already voted person");
        require(
        info[_address].authorized,
        "You Have No Right for Vote");
        info[_address].voted = true;
        teamA++;
        totalVotes++;
}

// Defining a function to check
// and skip the code if the person
// is already voted else allow to vote
// and calculate totalvotes for team B
function temaBF(address _address) public {
require(
!info[_address].voted,
"already voted person");
        require(
        info[_address].authorized,
        "You Have No Right for
        Vote"); teamB++;
        info[_address].voted = true;
        totalVotes++;
```

```
}

// Defining a function to check
// and skip the code if the person
// is already voted else allow to vote
// and calculate totalvotes for team C
function temaCF(address _address) public returns(
string memory){
        require(
        !info[_address].voted,
        "already voted person");
        require(
        info[_address].authorized,
        "You Have No Right for Vote");
        info[_address].voted = true;
        teamC++;
        totalVotes++;
        return("Thanks for Voting");
}

function totalVotesF() public view returns(uint){
        return totalVotes;
}

// Defining a function to announce
// the result of voting and
// the name of the winning team
function resultOfVoting() public view returns(
```

```
        string memory){
                if(teamA>teamB){
                        if(teamA>teamC){
                                return"A is Winning";
                        }
                        else if(teamC>teamA){
                                return "C is Winning"; } }
                else if(teamB>teamC) {
                        return "B is Winning";
                }
                else if(
                teamA==teamB && teamA==teamC || teamB==teamC ){
                        return "No One is Winning";
                }
        }
        }
```

```
class LoginPage extends StatelessWidget {
  Future<FirebaseApp> _initializeFirebase() async {
  FirebaseApp firebaseApp = await Firebase.initializeApp();
  return firebaseApp;
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

```
      title: Text('Firebase Authentication'),
    ),
    body: FutureBuilder(
      future: _initializeFirebase(),
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.done) {
          return Column(
            children: [
              Text('Login'),
            ],
          );
        }
        return Center(
          child: CircularProgressIndicator(),
        );
      },
    ),
  );
 }
}

class FireAuth {
  static Future<User?> registerUsingEmailPassword({
    required String name,
    required String email,
    required String password,
  }) async {
    FirebaseAuth auth = FirebaseAuth.instance;
```

```
    User? user;
    try {
      UserCredential userCredential = await auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      user = userCredential.user;
      await user!.updateProfile(displayName: name);
      await user.reload();
      user = auth.currentUser;
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        print('The password provided is too weak.');
      } else if (e.code == 'email-already-in-use') {
        print('The account already exists for that email.');
      }
    } catch (e) {
      print(e);
    }
    return user;
  }
}
```

```
import 'package:flutter/material.dart';
import  'package:flutter/services.dart';
import 'package:http/http.dart';
import 'package:web3dart/web3dart.dart';
```

```
class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  late Client httpClient;

  late Web3Client ethClient;

  final String myAddress = "0x8fF1b659bDC9D6eF5d99823B155cfdf47eF2944d";
  final String blockchainUrl =
      "https://rinkeby.infura.io/v3/4e577288c5b24f17a04beab17cf9c959";

  var totalVotesA;
  var totalVotesB;

  @override
  void initState() {
    httpClient = Client();
    ethClient = Web3Client(blockchainUrl, httpClient);
    getTotalVotes();
    super.initState();
  }

  Future<DeployedContract> getContract() async {
```

```
  String abiFile = await rootBundle.loadString("assets/contract.json");
  String contractAddress = "0x2D7870622599603625441644a66764cB08ac23D";
  final contract = DeployedContract(ContractAbi.fromJson(abiFile, "Voting"),
      EthereumAddress.fromHex(contractAddress));

  return contract;
}

Future<List<dynamic>> callFunction(String name) async {
  final contract = await getContract();
  final function = contract.function(name);
  final result = await ethClient
      .call(contract: contract, function: function, params: []);
  return result;
}

Future<void> getTotalVotes() async {
  List<dynamic> resultsA = await callFunction("getTotalVotesAlpha");
  List<dynamic> resultsB = await callFunction("getTotalVotesBeta");
  totalVotesA = resultsA[0];
  totalVotesB = resultsB[0];

  setState(() {});
}

snackBar({String? label}) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
```

```
      content: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Text(label!),
          CircularProgressIndicator(
          color: Colors.white,
          )
        ],
      ),
      duration: Duration(days: 1),
      backgroundColor: Colors.blue,
    ),
  );
}

Future<void> vote(bool voteAlpha) async {
  snackBar(label: "Recording vote");
  //obtain private key for write operation
  Credentials key = EthPrivateKey.fromHex(
      "f6417d3d4c5cc294ace85aa196fcde0ca792550e085f65fff459423e597ff306");

  //obtain our contract from abi in json file
  final contract = await getContract();

  // extract function from json file
  final function = contract.function(
    voteAlpha ? "voteAlpha" : "voteBeta",
  );
```

```
  //send transaction using the our private key, function and contract
  await ethClient.sendTransaction(
     key,
     Transaction.callContract(
        contract: contract, function: function, parameters: []),
     chainId: 4);
  ScaffoldMessenger.of(context).removeCurrentSnackBar();
  snackBar(label: "verifying vote");
  //set a 20 seconds delay to allow the transaction to be verified before trying to retrieve the balance
  Future.delayed(const Duration(seconds: 20), () {
  ScaffoldMessenger.of(context).removeCurrentSnackBar();
    snackBar(label: "retrieving votes");
    getTotalVotes();

    ScaffoldMessenger.of(context).clearSnackBars();
  });
}

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        padding: EdgeInsets.all(20),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
```

```
Container(
  padding: EdgeInsets.all(30),
  alignment: Alignment.center,
  decoration: const BoxDecoration(
     color: Colors.blue,
     borderRadius: BorderRadius.all(Radius.circular(20))),
  child: Row(
   mainAxisAlignment: MainAxisAlignment.spaceAround,
   children: [
     Column(
       children: [
       CircleAvatar(
          child: Text("A"),
        ),
        SizedBox(
          height: 10,
        ),
        Text(
          "Total Votes: ${totalVotesA ?? ""}",
          style: TextStyle(
              color: Colors.white, fontWeight: FontWeight.bold),
        )
       ],
     ),
     Column(
       children: [
       CircleAvatar(
          child: Text("B"),
```

```
        ),
        SizedBox(
          height: 10,
        ),
        Text("Total Votes: ${totalVotesB ?? ""}",
            style: TextStyle(
                color: Colors.white,
                fontWeight: FontWeight.bold))
      ],
    ),
  ],
 ),
),
SizedBox(
 height: 30,
),
Row(
 mainAxisAlignment: MainAxisAlignment.spaceAround,
 children: [
  ElevatedButton(
   onPressed: () {
   vote(true);
   },
   child: Text('Vote Alpha'),
   style: ElevatedButton.styleFrom(shape: StadiumBorder()),
  ),
  SizedBox(
   height: 30,
```

```
              ),
              ElevatedButton(
                onPressed: () {
                vote(false);
                },
                child: Text('Vote Beta'),
                style: ElevatedButton.styleFrom(shape: StadiumBorder()),
              )
            ],
          )
        ],
      ),
    ),
  );
 }
}

import 'dart:io';

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:url_launcher/url_launcher.dart';

class MyColors {
  Color color1 = Color(0xffE14D2A);
  Color color2 = Color(0xffFD841F);
  Color color3 = Color(0xffF58B65);
```

```
  Color color4 = Color(0xff001253);
  Color bottomBarIconColor = Colors.black45;
  Color bottomBarSelectedIconColor = Colors.orange;
}



Future<bool> isConnected() async {
  final result = await InternetAddress.lookup('example.com');
  if (result.isNotEmpty && result[0].rawAddress.isNotEmpty) {
    print(true);
    return true;
  } else {
    print(false);
    return false;
  }
}

class Styles {
  TextStyle questionStyle =
      GoogleFonts.roboto(fontSize: 18, fontWeight: FontWeight.w800);
  TextStyle subQuestionStyle = GoogleFonts.roboto(fontSize: 16);
  TextStyle optionsStyle =
      GoogleFonts.roboto(fontSize: 16, fontWeight: FontWeight.w400);
  TextStyle mainTitle =
      GoogleFonts.roboto(fontSize: 20, fontWeight: FontWeight.bold);
  TextStyle titleStyle =
      GoogleFonts.roboto(fontSize: 18, fontWeight: FontWeight.bold);
```

```
  TextStyle bodyStyle = GoogleFonts.roboto(fontSize: 14);
}

bool isValid(String pattern,String match) {
    // Pattern pattern =
    // r+";
    RegExp regex = new RegExp(pattern);
    return regex.hasMatch(match);
  }

library globalWidgets;

import 'package:flutter/material.dart';

AppBar appBar() {
  return AppBar(
  centerTitle: true,
  title: Container(
      width: 130,
      child: Image.asset(
        'assets/images/logo.png',
        fit: BoxFit.fitWidth,
      ),
    ),
    elevation: 0,
    backgroundColor: Colors.white,
    iconTheme: IconThemeData(color: Colors.black),
  );
```

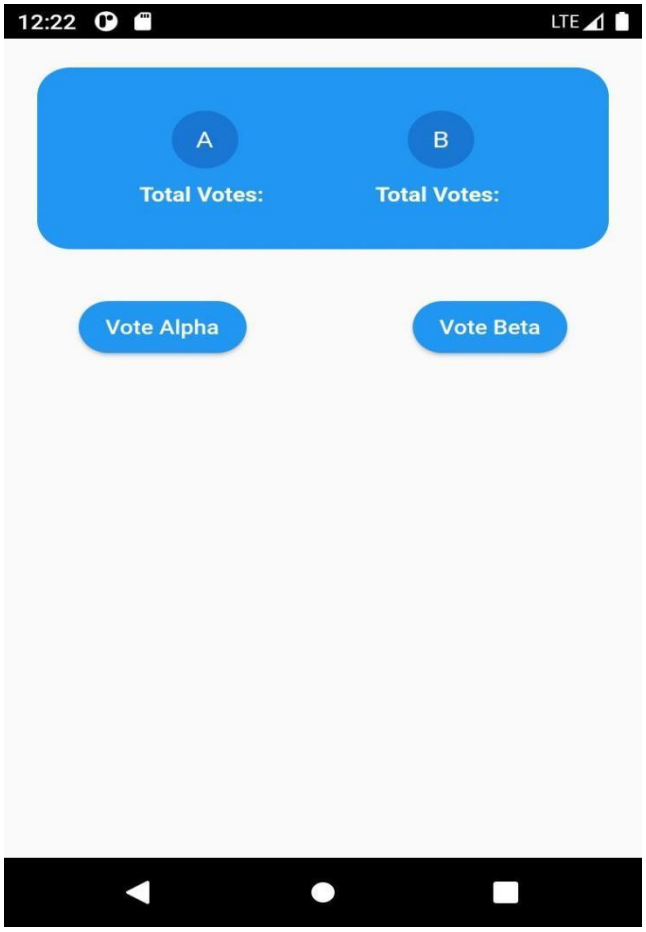
```
}

Drawer drawer() {
  return Drawer(
  child: ListView(
     padding: EdgeInsets.zero,
     children: <Widget>[
     SizedBox(
        height: 150,
        child: Padding(
          padding: const EdgeInsets.all(8.0),
          child: ClipRRect(
            borderRadius: BorderRadius.all(Radius.circular(16.0)),
            child: DrawerHeader(
              // decoration: BoxDecoration(
              //     border: Border.all(
              //         // color: Colors.red[500],
              //         ),
              //     borderRadius:
              BorderRadius.all(Radius.circular(20))), child: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text('K Vijay Kumar'),
                    Text('1-Mar-2022'),
                    Text('17.385N, 78.487E')
                  ],
```

```
          ),
        ),
        decoration: BoxDecoration(
          color: Color(0xff3E6D9C),
        ),
      ),
    ),
  ),
),
Center(
  child: SizedBox(
    // width: 100,
    // height: 100,
    child: Card(
      color: Color(0xffE14D2A),
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Center(child: Text('Targated Surveys')),
      ),
    ),
  ),
),
Center(
  child: SizedBox(
    // width: 100,
    // height: 100,
    child: Card(
      color: Color(0xffE14D2A),
```

```
            child: Padding(
              padding: const EdgeInsets.all(8.0),
              child: Center(child: Text('Non-Targated Surveys')),
            ),
          ),
        ),
      )
    ],
  ),
 );
}
```

Fontend for sample project:



CHAPTER- 4
Conclusion

With a few recommendations for best practices in the future, this section assesses and concludes the research study presented in this piece[11]. However, there are a few limitations that can be overcome in the further applications of this work and useful enhancement recommendations. Significant voting system uncertainties threaten the stability of several countries.We created a blockchain-based, smart contract-based digital voting system to ensure voter legitimacy, fairness of polling data, and non-manipulative vote counting. Three smart contracts are used in this system to carry out various activities related to the entire election process. As a result, the third party's engagement is lower than it is for other contemporary systems. The results of the votes cast are kept secret until after the election. Who cast the ballot has not been made public. To prevent anyone in the network from being able to identify the voter, we have stored the voter's information as a hash.

The data is saved in this case as a hash rather than its complete form, which lowers expense as well. Voters can also use their vote ID, which they receive while casting their vote, to confirm their vote after the election has ended. With the use of smart devices and this process, voters can cast their ballots for the candidate they want from anywhere in the world. By increasing the number of voters, this would contribute to the spread of democracy worldwide. Given the greatest levels of security our technology offers, including anonymity, integrity, security, privacy, fairness, verifiability, and mobility, it can

be successfully used in the electoral process.

This method's drawback is that we don't offer the OTP (One Time Password) option upon registration. Another problem is that during the voting process, we kept the encrypted vote in the blockchain. Once the election has concluded, this data won't be used again.The previously described restrictions will be addressed in subsequent development[19].

CHAPTER-5
References

1. Haibo Yi, Securing e-voting based on blockchain in P2P network.Yi Eurasip Journal (2019)

2. Jack Ahlkvist, Anton Gustafsson, Carl Lundborg, Joakim Mattsson Thorell, Aron Sandstedt, Sanjin Slavnic, A Decentralized Voting System, DeVote. Chalmers University of Technology (2019)

3. Y. Xie, Who Overreports Voting?J J. Am. Polit. Sci. Rev. (2017)

4.. T. Green, O. Sarrasin, R. Baur, et al., From stigmatized immigrants to radical right voting: a multilevel study on the role of threat and contact[J]. Polit. Psychol. (2016)

5. B. Los, P. Mccann, J. Springford, et al., The mismatch between local voting and the local economic consequences of Brexit[J]. Reg. Stud. (2017)

6. S. Eijffinger, R. Mahieu, L. Raes, Inferring hawks and doves from voting records[J]. Eur. J. Polit. Econ. Elsevier. (2017)

7. Ethereum community. Ethereum Introduction. [Online; accessed 7-May-2019]. [Online].
Available: https://github.com/ethereum/wiki/wiki/Ethereum-introduction

8. C. Dannen, *Smart Contracts and Tokens*. Berkeley, CA: Apress, 2017, pp. 89–110. [Online]. Available: https://doi.org/10.1007/978-1-4842-2535-6 5

9. ——, *Solidity Programming*. Berkeley, CA: Apress, 2017, pp. 69–88. [Online]. Available: https://doi.org/10.1007/978-1-4842-2535-6 4

10. Contracts - Solidity 0.5.5 documentation. [Accessed 30-April-2019]. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.5/contracts.html

11. T. suite, “Drizzle,” 2019. [Online]. Available: https://trufflfflffleframework.com/drizzle

12. Rajendran Anandha Jothi. Confidential e-voting system using face detection and recognition. 3, 06 2018.

13. B. Rogers, S. Chhabra, M. Prvulovic, and Y. Solihin. Using address independent seed encryption and bonsai merkle trees to make secure processors os- and performance-friendly. In 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007), pages 183–196, 2007.

14. Chang-Hyun Roh* and Im-Yeong Lee**. A study on electronic voting system using private blockchain. JOURNAL OF INFORMATION PROCESSING SYSTEMS, 16, 04, 2020.

15. Seifelnasr, Mohamed, Galal, Hisham S., Youssef, Amr M., 2020. Scalable open-vote network on ethereum. IACR Cryptology ePrint Archive 2020, 33.

16. Shahzad, B., Crowcroft, J., 2019. Trustworthy electronic voting using adjusted blockchain technology. IEEE Access 7, 24477–24488.

17. Samarth Shakya and Vivek Kapoor. A decentralized polling system using ethereum technology. Journal of Information Technology Management, 14 (Security and Resource Management challenges for Internet of Things): 1–8, 2022.

18. Sheer Hardwick, F., Gioulis, A., Naeem Akram, R., Markantonakis, K., 2018. E-voting with blockchain: An e-voting protocol with decentralization and voter privacy. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1561–1567.

19. Sourav Rajeev Rohan Varghese Mathew V. Arun, Aditya Dutta. E-voting using a decentralized ethereum application. 04 2019.

20. Sudhanshu Saxena Dr. Hemavathi P Ananya Sumit Kumar, Darshini N. Voteeth: An e-voting system using blockchain. In International Research Journal of Computer Science (IRJCS), volume 5, 2019.

.