Experiment No.:9

PART A

Online Voting System Using Blockchain

1. Abstract:

The **Online Voting System Using Blockchain** project aims to revolutionize traditional voting systems, addressing prevalent issues like fraud, tampering, inefficiencies, and high operational costs. Traditional electoral processes, especially in large democracies like India, are resource-intensive and vulnerable to tampering. Blockchain, with its decentralized and immutable characteristics, offers a solution by securely recording votes in a tamper-proof manner. This system leverages **Ethereum blockchain** to ensure the integrity, security, and transparency of the voting process, allowing votes to be stored immutably and auditable for transparency.

One of the core challenges with blockchain-based systems is **scalability**. Blockchain transactions can be costly and time-consuming when applied to large-scale elections, such as those conducted in India. To mitigate this, the project adopts a **hybrid approach**, utilizing **Firebase** as a **secondary database** to manage non-sensitive data like voter profiles, logs, and election metadata. This off-chain storage reduces the transaction load on the blockchain, enabling faster performance and lower operational costs, while the blockchain itself securely handles the critical aspects of vote recording and result declaration.

The system also incorporates **multi-factor authentication (MFA)** and **biometric verification**, ensuring that only authorized voters can access the platform. **MFA**, combined with **biometric data** (fingerprint or facial recognition), adds an additional layer of security, guaranteeing the legitimacy of the voter and protecting the voting process from unauthorized access. Once authenticated, voters can securely cast their votes through the mobile app, and their votes are immediately recorded on the blockchain.

The **real-time result visualization** feature is another key component of the system. As votes are recorded on the blockchain, results are processed and displayed in real-time, providing an immediate and transparent overview of the election's progress. This

feature supports region-wise classification and detailed result breakdowns for elections ranging from **Gram Panchayat** to **Lok Sabha**, ensuring that voters and election officials have full transparency throughout the election cycle. The blockchain guarantees that votes cannot be modified once cast, eliminating any risk of post-election tampering or manipulation.

To ensure voter privacy and anonymity, the system isolates sensitive voting data on the blockchain, while Firebase handles metadata like whether a voter has participated. This separation ensures that voter data is never exposed, yet it allows election officials to verify voter participation without revealing sensitive information.

The project successfully demonstrates a scalable, cost-effective, and **secure alternative to traditional voting systems**. By eliminating the need for extensive manpower and reducing reliance on physical infrastructure, such as Electronic Voting Machines (EVMs), the blockchain-based system offers a modernized approach to conducting elections, with a focus on security, transparency, and efficiency. Its real-time results and auditability enhance voter trust, making the system particularly suitable for large-scale elections in India and globally.

The proposed system ensures that elections are **tamper-proof**, **transparent**, and **cost-efficient**, offering a blueprint for modernizing the electoral process through blockchain technology.

2. Introduction:

The electoral process is a cornerstone of democracy, providing citizens the means to participate in governance. In large democratic countries like India, where elections are held on a vast scale, traditional voting methods, such as paper ballots and **Electronic Voting Machines (EVMs)**, face significant challenges. These challenges include **voter fraud**, **vote tampering**, **inefficiency**, and **high operational costs**. The centralized nature of these systems makes them vulnerable to manipulation, raising concerns about the integrity of election results and eroding public trust in the electoral process.

In recent years, **blockchain technology** has emerged as a robust solution to these problems. Blockchain's **decentralized, immutable, and transparent** nature offers the potential to revolutionize electoral systems by providing a secure, efficient, and tamper-

proof method of recording votes. By leveraging blockchain, each vote can be stored immutably on a decentralized ledger, ensuring that once a vote is cast, it cannot be altered or deleted. This enhances voter confidence and eliminates the risks associated with traditional systems. Moreover, blockchain's ability to automate processes through **smart contracts** significantly reduces the need for human intervention, minimizing errors and ensuring timely vote counting and result declaration.

This project aims to design and implement a **Blockchain-Based Online Voting System** that integrates the security of blockchain with the accessibility and ease of a mobile voting application. The system addresses the scalability issues of blockchain by incorporating a **secondary database (Firebase)** to manage non-sensitive data, such as voter profiles and election logs, optimizing the system's performance for large-scale elections. The project also prioritizes **security and privacy**, ensuring that voters can cast their votes securely and anonymously while election officials can audit results transparently.

Through this system, the project seeks to modernize the electoral process, providing a **secure**, **cost-effective**, and **transparent** alternative to traditional voting methods. This is particularly relevant in the context of India, where elections involve millions of voters across diverse regions and scales, from **Gram Panchayat** to **Lok Sabha** elections. The proposed system ensures that the integrity of the voting process is maintained while significantly reducing the resources, time, and costs required to conduct elections.

3. Methodology:

This chapter outlines the methodology adopted in the development of the **Blockchain-Based Online Voting System**. The methodology encompasses the tools and technologies used, the process flow of the system, and the steps taken to ensure security, performance, and user experience.

3.1 Blockchain Integration

3.1.1 Development Environment:

- **Ganache/Truffle:**

  During the development phase, Ganache and Truffle will be used to simulate a local Ethereum blockchain environment. This will enable

testing of smart contracts and voting transactions in a controlled setup before deploying to the live Ethereumnetwork.

- **Web3.js and Node.js:**

  These tools will facilitate communication between the mobile app, backend, and blockchain. Web3.js will handle smart contract interactions from the frontend, while Node.js will manage API requests and backend operations.

3.1.2 Voting Process:

- **Vote Transactions:**

  Once a voter casts a vote, the mobile app sends the transaction to the backend. The backend then interacts with the Ethereum blockchain using smart contracts to record each vote as a transaction.

- **Vote Validation:**

  Smart contracts written in Solidity will validate each vote, ensuring authenticity and preventing multiple votes from the same voter.

- **Result Publication:**

  As soon as voting is concluded, smart contracts automatically count votes and publish results in real-time on the blockchain, providing transparency and immutability.

3.2 Security and Privacy

3.2.1 Multi-factor Authentication (MFA):

The system will incorporate MFA, including biometric or OTP-based authentication, toensure that only legitimate users can vote.

3.2.2 End-to-End Encryption:

All data transmitted between the mobile app, backend, and blockchain is encrypted toprotect user privacy and system security.

3.2.3 Anonymity and Transparency:

- **Blockchain Ledger:**

  While the blockchain maintains a transparent ledger of transactions (votes), the anonymity of each vote is preserved, ensuring voter privacy.

- **Smart Contracts for Security:**

  Smart contracts ensure that each vote is unique, correctly counted, and tamper-proof.

3.3 Performance Optimization

3.3.1 Data Segmentation:

- **Blockchain for Votes:**

  Only sensitive data, such as votes and election results, are stored on the Ethereum blockchain to leverage its immutability and security features.

- **Secondary Database for Non-sensitive Data:**

  Non-vote-related data, such as voter profiles, election metadata, and user activitylogs, will be stored in secondary database. This approach reduces the load on the blockchain and ensures faster system performance, particularly during peak voting times.

3.3.2 Cost-Effectiveness:

- **Transaction Batching:**

  To minimize gas fees and improve performance, vote transactions are batchedtogether during high voting volumes.

- **Optimized Scalability:**

  By offloading non-critical data to secondary database, the system remains scalable and cost-effective during large-scale elections, such as national elections.

3.4 Process Breakdown

3.4.1 Voter Registration:

- **User Flow:**

  Users register through the mobile app by providing identification details, which are verified through MFA. Once verified, the system generates a unique voter ID, which is stored on the blockchain for future voting authentication.

3.4.2 Vote Casting:

- **User Flow:**

  After selecting a candidate from the list of eligible elections, the vote is securely transmitted to the Ethereum blockchain via the backend. A confirmation message is displayed in the app once the vote is successfully recorded on the blockchain.

- **Transaction Validation:**

  Smart contracts verify the legitimacy of the vote and ensure that no voter casts more than one vote.

3.4.3 Data Handling and Storage:

- **Sensitive Data:**

  Votes are stored immutably on the blockchain, ensuring that they cannot be altered ortampered with once submitted.

- **Non-Sensitive Data**

  Voter profiles, election logs, and other non-critical information are stored in secondary database, optimizing the system's speed and reducing blockchaincongestion.

3.4.4 Result Display:

- **Real-Time Updates:**

  Once voting is complete, the system uses smart contracts to process and count votes. The results are automatically updated and displayed in real-time through the mobile app, with both list and graphical formats available for users.

- **Transparency:**

  Since votes are recorded on the blockchain, the result calculation process is transparent and auditable, enhancing the election's credibility.

Blockchain-Based Voting Security:

The system draws on research, such as the secure electronic voting systems outlined [2], which emphasize the use of blockchain and mobile technologies for privacy and security in elections.

Scalability Insights:

The system's architecture takes lessons [16], which addresses scalability challenges in blockchain- based applications, ensuring that the voting system can handle high voter turnout without compromising performance.

The methodology combines cutting-edge blockchain technology, a secure backend, and optimized datahandling using secondary database to build a robust, scalable, and secure online voting system. The useof smart contracts, MFA, and real-time results ensures the system's transparency and trustworthiness, while performance optimizations make it feasible for large-scale elections globally.

4. Technology Used to develop and Deploy application:

4.1 HARDWARE USED:

4.1.1 Mobile Devices:

- Smartphones (iOS and Android)
- Tablets (iOS and Android)

4.1.2 Servers:

- Blockchain nodes servers
- Backend servers

4.1.3 Cryptographic Hardware:

- Hardware Security Modules (HSM)

4.1.4 Development and Testing Hardware:

- Laptops/Desktops for developers
- Test devices (various models of smartphones and tablets)

4.1.5 Security Hardware:

- Biometric authentication devices (optional, for enhanced security)

4.2 SOFTWARE USED:

4.2.1 Blockchain Network:

- Ethereum

4.2.2 Smart Contract (Development):

- Solidity

4.2.3 Mobile App Development Frameworks:

- Android
- Flutter

4.2.4 Database:

- Firebase
- MongoDB

4.2.5 Development and Testing Tools:

- Truffle
- Ganache
- Android Studio
- Remix IDE

5. Implementation Details:

The implementation of the Online Voting System using blockchain technology is centered around the development of a secure, scalable, and efficient voting platform. Below is an overview of the system's key implementation features:

- **Smart Contracts**:

  Smart contracts written in Solidity are used to manage voter authentication, vote validation, and real-time vote counting. The contract is deployed on the Ethereum blockchain using Ganache for local testing and Truffle for managing the contracts.

- **Mobile Application**:

  The mobile app, developed using Android Studio, serves as the front-end through which voters can register, authenticate, and cast their votes. It communicates with the backend via Web3.js for blockchain interactions.

- **Backend & Database**:

  The system uses Node.js for backend operations, managing requests from the app and interacting with the blockchain. A secondary database (Firebase) stores non-sensitive information like voter profiles, while the blockchain handles the vote data.

- **Multi-Factor Authentication (MFA)**:

  For enhanced security, MFA using biometrics or OTP is integrated, ensuring only legitimate users can vote.

- **Real-Time Results**:

  Once the voting period concludes, smart contracts count and display results in real time. The data is presented in textual and graphical formats via the mobile app, ensuring transparency and ease of access for election officials.

6. Output:

The **Online Voting System Using Blockchain** is expected to deliver the following key outcomes:

6.1 Secure Voting Process:

- **Immutable Vote Recording:**

  Votes cast through the mobile app will be stored immutably on the Ethereum blockchain, ensuring that they cannot be tampered with or altered after submission. This guarantees the integrity of the voting process.

6.2 Real-Time Results:

- **Transparency and Speed:**

  The system will display election results in real-time, with both textual and graphical representations of the number of votes each candidate has received. This feature will enhance transparency, allowing both voters and election officials to monitor results as they are tallied.

6.3 Auditability:

- **Comprehensive Election Logs:**

  Election-related data, such as voter participation, metadata, and activity logs, will be stored in secondary database, enabling election officials and auditors to review the process without compromising voter anonymity. This ensures a transparent and auditable electoral process.

6.4 Cost Efficiency:

- **Reduction of Election Costs:**

  By leveraging blockchain technology and reducing the need for physical voting machinery like Electronic Voting Machines (EVMs), the system significantly cuts costs associated with running elections. Additionally, it minimizes manpower requirements, making it more cost-effective for large-scale national elections.

6.5 Scalability and Performance:

- **Handling Large Voter Bases:**

  The system is designed to handle high user loads during peak election periods, such as national elections. The use of transaction batching and offloading non-sensitive data to secondary database ensures that the blockchain remains scalable and efficient.

6.6 Enhanced Security:

- **Multi-Factor Authentication (MFA):**

  The system ensures that only verified voters can access the platform and cast votes,adding an extra layer of security to prevent unauthorized access.

- **End-to-End Encryption:**

  All communications between the mobile app, backend, and blockchain are encrypted, further enhancing the security of voter data and the election process.

6.7 User Engagement and Voter Participation:

- **Participation Tracking:**

  The system provides features that allow election officials to track voter participation in real-time. This information can be used to improve voter turnout and identify patterns in voter behavior.

The system is expected to deliver a secure, transparent, and cost-effective solution for conducting elections,capable of scaling for both local and national elections in India and globally. By leveraging blockchain technology for vote storage and using secondary database for additional data management, the system ensures performance, security, and transparency while maintaining voter privacy.

7. Conclusion:

The **Blockchain-Based Online Voting System** developed in this project represents a significant advancement in the electoral process, particularly in large-scale scenarios such as those encountered in India. The integration of **Ethereum's blockchain technology** with a **secondary database** creates a secure, scalable, and efficient platform for conducting elections.

By utilizing blockchain, the system ensures that votes are recorded immutably, providing a robust mechanism to prevent fraud and vote tampering. The decision to incorporate a secondary database, such as **Firebase**, allows for the effective handling of non-sensitive data, optimizing system performance and reducing operational costs. This hybrid approach addresses critical challenges associated with traditional voting systems, including security vulnerabilities, inefficiencies, and high costs.

The proposed system effectively tackles issues of scalability through techniques like **transaction batching** and the potential implementation of **Layer-2 solutions**, ensuring it can accommodate a high volume of votes during peak times. Furthermore, the use of **end-to-end encryption**, **multi-factor authentication (MFA)**, and comprehensive **audit logs** enhances the overall security of the voting process. Real-time result processing and transparent auditing mechanisms instill greater confidence among voters, reinforcing trust in the electoral system.

In summary, the blockchain-based voting system presents a transformative step toward modernizing elections, offering a viable and trustworthy solution for secure, scalable, and efficient democratic processes. By minimizing reliance on manual processes and traditional voting machines, this system significantly contributes to the evolution of electoral systems in the digital age.

REFERENCES:

1. D.Dwijesh Kumar, D.V. Chandini, Dinesh Reddy, "Secure Electronic Voting System using Blockchain Technology", International Journal of Smart Home, **(2020)**

2. Wenbin Zhang, Sheng Huang, Yuan Yuan, Yanyan Hu, Shaohua Huang, Shengjiao Cao, Anuj Chopra, "A Privacy-Preserving Voting Protocol on Blockchain", IEEE 11th International Conference on Cloud Computing, **(2018)**

3. G.Kalaiyarasi, T.Narmadha, K. Balaji, V.Naveen, "E-Voting System In Smart Phone Using Mobile Application", 6th International Conferenceon Advanced Computing & Communication System (ICACCS), **(2020)**

4. Stephan Neumann, Oksana Kulyk, Melanie Volkamer, "A Usable Android Application Implementing Distributed Cryptography For Election Authorities", 9th International Conference on Availability, Reliability and Security, **(2014)**

5. Jae-Geun Song, Sung-Jun Moon, Ju-Wook Jang, "A Scalable Implementation of Anonymous Voting over Ethereum Blockchain", Sensors 21, no. 12 **(2021)**: 3958.

6. Akhil Shah, Nishita Sodhia, Shruti Saha, Soumi Banerjee, Madhuri Chavan, "Blockchain Enabled Online-Voting System", **(2020)**

7. Cristian Toma, Marius Popa, Catalin Boja, Cristian Ciurea, Mihai Doinea, "Secure and Anonymous Voting D-App with IoT Embedded Device Using Blockchain Technology", Electronics, 11(12), 1895., **(2022)**

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

Roll. No.: 28	Name: Harsh A. Minde.
Class: BE COMP C	Batch: C2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student:

```
import 'package:flutter/services.dart';
import 'package:voting_dapp/utils/constants.dart';
import 'package:web3dart/web3dart.dart';

Future<DeployedContract> loadContract() async {
  String abi = await rootBundle.loadString('assets/abi.json');
  String contractAddress = contractAddress1;
  final contract = DeployedContract(ContractAbi.fromJson(abi, 'Election'),
      EthereumAddress.fromHex(contractAddress));
  return contract;
}

Future<String> callFunction(String funcname, List<dynamic> args,
    Web3Client ethClient, String privateKey) async {
  EthPrivateKey credentials = EthPrivateKey.fromHex(privateKey);
  DeployedContract contract = await loadContract();
  final ethFunction = contract.function(funcname);
  final result = await ethClient.sendTransaction(
      credentials,
```

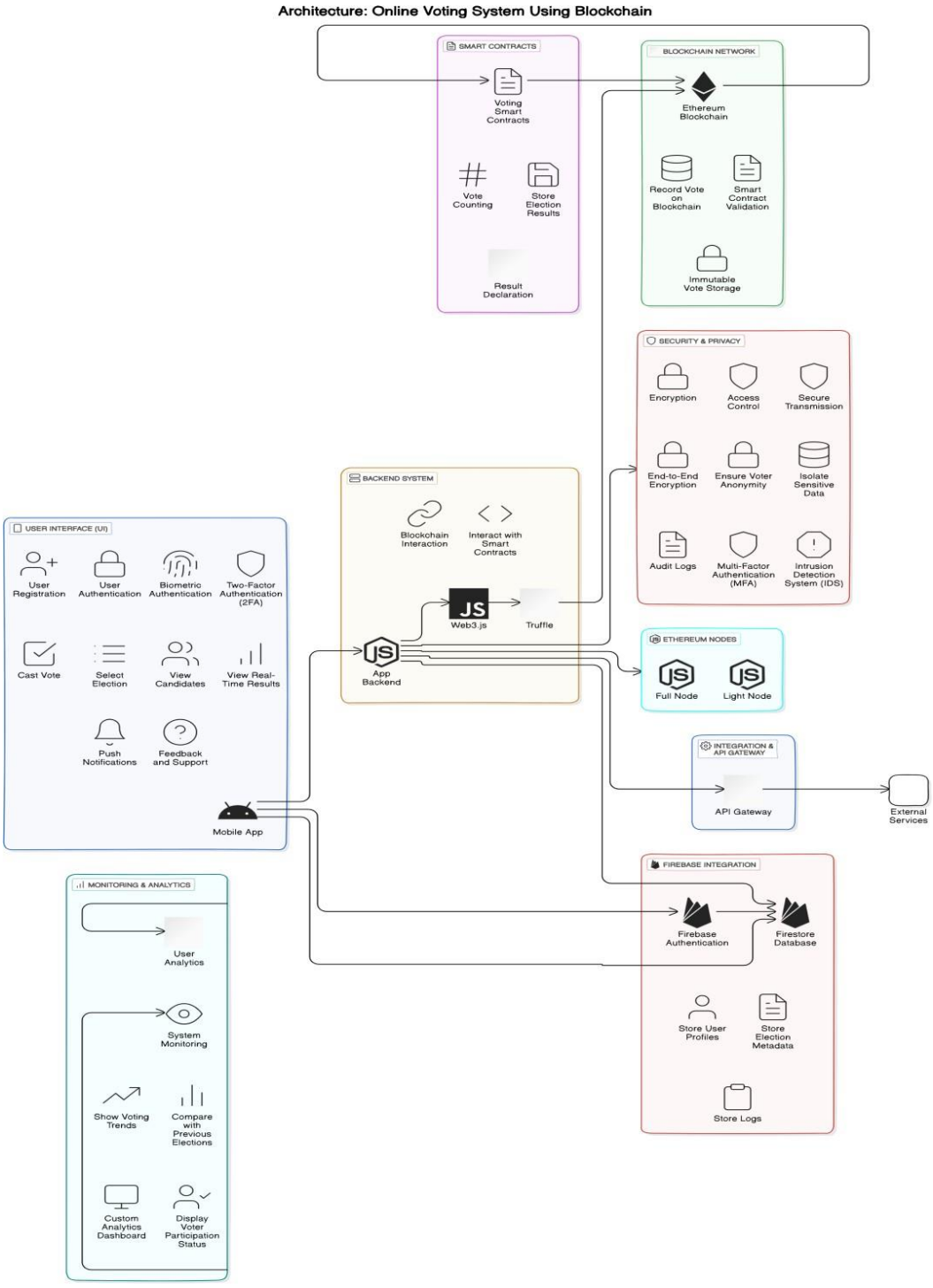
```
    Transaction.callContract(
      contract: contract,
      function: ethFunction,
      parameters: args,
    ),
    chainId: null,
    fetchChainIdFromNetworkId: true);
 return result;
}

Future<String> startElection(String name, Web3Client ethClient) async {
 var response =
    await callFunction('startElection', [name], ethClient, owner_private_key);
 print('Election started successfully');
 return response;
}

Future<String> addCandidate(String name, Web3Client ethClient) async {
 var response =
    await callFunction('addCandidate', [name], ethClient, owner_private_key);
 print('Candidate added successfully');
 return response;
}

Future<String> authorizeVoter(String address, Web3Client ethClient) async {
 var response = await callFunction('authorizeVoter',
    [EthereumAddress.fromHex(address)], ethClient, owner_private_key);
 print('Voter Authorized successfully');
 return response;
}
```

```
Future<List> getCandidatesNum(Web3Client ethClient) async {
  List<dynamic> result = await ask('getNumCandidates', [], ethClient);
  return result;
}

Future<List> getTotalVotes(Web3Client ethClient) async {
  List<dynamic> result = await ask('getTotalVotes', [], ethClient);
  return result;
}

Future<List> candidateInfo(int index, Web3Client ethClient) async {
  List<dynamic> result =
      await ask('candidateInfo', [BigInt.from(index)], ethClient);
  return result;
}

Future<List<dynamic>> ask(
    String funcName, List<dynamic> args, Web3Client ethClient) async {
  final contract = await loadContract();
  final ethFunction = contract.function(funcName);
  final result =
      ethClient.call(contract: contract, function: ethFunction, params: args);
  return result;
}

Future<String> vote(int candidateIndex, Web3Client ethClient) async {
  var response = await callFunction(
      "vote", [BigInt.from(candidateIndex)], ethClient, voter_private_key);
  print("Vote counted successfully");
```

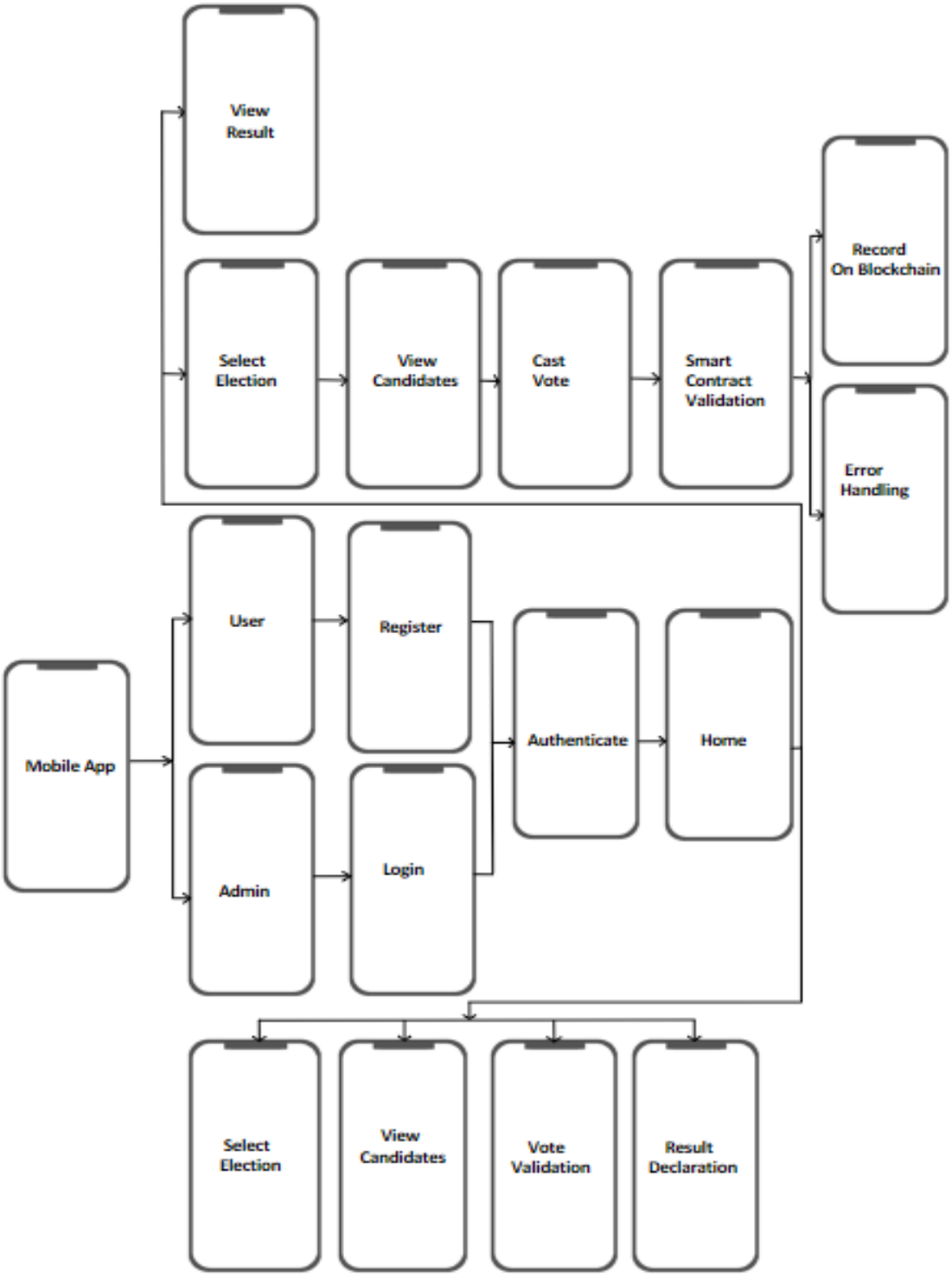
```
return response;
}
```

B.2 Input and Output:

1) Architecture of the system:



2) Wire-Frame Design:



B.3 Observations and learning:

1. **Efficient Blockchain Interactions**:

   The smart contract functions interact seamlessly with the Ethereum blockchain using Web3dart. Each transaction is securely recorded on the blockchain, ensuring immutability and transparency.

2. **Real-Time Transaction Feedback**:

   After performing voting actions, the system provides immediate feedback. The transaction hash confirms that the vote is securely recorded on the blockchain, reinforcing the decentralized nature of the election.

3. **Data Security and Privacy**:

   The use of blockchain ensures that vote tampering is impossible. Votes are stored in a decentralized ledger, maintaining both security and transparency while ensuring voter privacy.

4. **User-Friendly Interface**:

   The mobile app provides a straightforward interface for managing elections, voting, and authorizing voters. This simplicity ensures that even users unfamiliar with blockchain technology can participate without difficulty.

5. **Learning Experience**:

   The project enhanced understanding of integrating Flutter with blockchain, specifically managing smart contracts through Web3dart. It highlighted the importance of secure transaction handling, decentralization, and real-time data validation in the context of elections.

B.4 Conclusion:

The implementation of the Online Voting System using Flutter and Web3dart demonstrates how blockchain technology can revolutionize traditional election systems. Through the immutability and transparency of the Ethereum blockchain, the system ensures that votes are securely recorded, tamper-proof, and transparent to both voters and officials.

Key accomplishments of this phase include:

- **Secure Voting Operations**: Elections are conducted transparently with each vote being immutably stored on the blockchain.
- **Ease of Use**: The mobile app offers a user-friendly interface for managing election tasks such as starting elections, adding candidates, authorizing voters, and casting votes.
- **Future Expansion**: The project provides a strong foundation for future features such as real-time result visualization, performance optimizations, and multi-factor authentication (MFA) for improved security.

Overall, this project lays the groundwork for a scalable and secure election system suitable for large-scale democratic processes, enhancing both trust and efficiency through the application of blockchain technology.

THANK YOU !!