RV COLLEGE OF ENGINEERING

BENGALURU- 560059

(Autonomous Institution affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

“Decentralized KYC verification”

BLOCKCHAIN TECHNOLOGY AND USE CASE (18IS7F2)

Experiential Learning VII Semester

Academic year 2023-2024

Submitted by

Kanupriya Anand (1RV20IS067)

Under the guidance of

Prof. Sharadadevi K

Assistant Professor, Dept of ISE

RV College of Engineering

RV COLLEGE OF ENGINEERING

BENGALURU- 560059

(Autonomous Institution affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the work titled **“Decentralized KYC verification”** has been carried out by **Kanupriya Anand** (1RV20IS067), bona fide student of RV College of Engineering, Bengaluru, who has submitted in partial fulfillment for the Assessment of Course: **Blockchain Technology and Use Case (18IS7F2) – Experiential Learning** during the year 2023-2024. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report.

Faculty in-charge	**Head of Department**
Prof Sharadadevi K	Dr Sagar B M

ABSTRACT

In the financial sector, the Know Your Customer (KYC) process is pivotal for regulatory compliance, fraud prevention, and anti-money laundering (AML) efforts. However, traditional KYC systems are fraught with challenges, including high operational costs, inefficiencies, data security vulnerabilities, and privacy concerns. This project proposes a novel solution to these issues through the development of a decentralized KYC system built on the Ethereum blockchain. Leveraging the inherent security, transparency, and immutability of blockchain technology, this system aims to enhance data security and privacy, streamline operational processes, and reduce associated costs.

The project employs Ethereum smart contracts to automate KYC verification processes, enabling a trustless and tamper-proof environment for the secure and efficient exchange of information. By decentralizing data storage, the system ensures that individuals retain control over their personal information, sharing it selectively and securely with financial institutions. This approach not only mitigates risks associated with data breaches and unauthorized access but also fosters a more customer-centric KYC process.

Furthermore, the project seeks to establish a standardized and interoperable KYC framework that can be adopted across various financial institutions and jurisdictions. This standardization promises to reduce redundancy and improve compliance efficiency, aligning with global regulatory standards and facilitating smoother cross-border financial transactions. Through this initiative, the practical application and benefits of blockchain technology is addressed by addressing longstanding challenges within the financial services industry. The decentralized Ethereum-based KYC system represents a significant step forward in the pursuit of more secure, efficient, and user-friendly financial processes, setting a precedent for future innovations in the sector.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1. **Understanding KYC:**
   - KYC stands for "Know Your Customer," a mandatory process for banks and financial institutions to verify the identity of their clients.
   - Essential for preventing fraud, money laundering, and terrorist financing.
2. **Challenges with Traditional KYC:**
   - **Centralization:** Traditional KYC processes are centralized, creating a single point of failure that hackers can target, leading to privacy breaches and data theft.
   - **Inefficiency and Cost:** Manual verification is time-consuming and expensive, requiring significant human resources and leading to customer dissatisfaction due to delays.
   - **Data Redundancy:** Each institution conducts its own KYC, causing unnecessary repetition and inconvenience to customers.
3. **Why Decentralize the KYC Process?**
   - **Enhanced Security:** A decentralized system reduces the risk of data breaches by distributing data across a network, rather than storing it in a central database.
   - **Privacy and Control:** Empowers users by giving them control over their personal information, sharing only what’s necessary, with whom they choose.
   - **Efficiency and Cost Reduction:** Streamlines the verification process through smart contracts on the Ethereum blockchain, significantly reducing the time and cost associated with KYC compliance.
   - **Interoperability:** Creates a unified and standardized process that can be used across multiple platforms and jurisdictions, eliminating redundancy and enhancing customer experience.
4. **The Role of Ethereum:**
   - Ethereum's blockchain offers a secure and transparent platform for deploying smart contracts that automate and enforce the KYC process, ensuring compliance and trust without intermediaries.

1.1 Topic relevance

Foundational Blockchain Concepts:

- **Decentralization:** This project illustrates the core principle of blockchain by moving away from centralized systems to a distributed ledger approach.
- **Consensus Mechanisms**: Demonstrates how transactions and data verifications are validated on the blockchain, ensuring integrity and security without a central authority.

Smart Contracts and Ethereum:

- **Smart Contracts Utilization:** Showcases how Ethereum's smart contracts automate processes, in this case, KYC verifications, making them more efficient and tamper-proof.
- **Ethereum's Flexibility:** Highlights Ethereum's capabilities beyond just transactions—supporting DApps (Decentralized Applications) that can transform industries.

Application in Financial Services:

- **Regulatory Compliance:** Addresses how blockchain can meet global KYC compliance needs more efficiently, reducing costs for financial institutions and improving the customer experience.
- **Privacy and Security:** Demonstrates solving critical industry challenges like data security and privacy through encryption and decentralized data storage.

1.2 Objectives

- To implement an Ethereum-based decentralized KYC system for identity verification
- To provide functionality for registering financial institutions like banks and their customers
- To utilize blockchain's inherent security features to ensure that personal information is encrypted and only accessible via permissioned access, thereby enhancing privacy and data protection.

Chapter 2

LITERATURE SURVEY

[1]The Know Your Customer (KYC) protocols have become a cornerstone in the global financial system, primarily aimed at combating money laundering and terrorist financing. Traditional KYC processes, however, are criticized for their inefficiency, high costs, and privacy concerns. Smith and Taylor (2020) provide an extensive overview of the evolution of KYC, noting the increasing regulatory burden on financial institutions and the consequent impact on customer onboarding processes.

[2]The advent of blockchain technology introduced a paradigm shift towards decentralization, offering a novel approach to data management and security. Lee (2019) explores the fundamental principles of blockchain, emphasizing its potential to transform traditional centralized systems into decentralized networks, enhancing transparency, security, and efficiency.

[3]Recent literature has begun to explore the application of blockchain technology in KYC processes. Anderson et al. (2021) discuss how blockchain can address inefficiencies in traditional KYC practices by enabling secure, immutable, and transparent verification processes. The study highlights the potential for cost reduction, improved customer experience, and enhanced data security.

[4]Ethereum's platform, known for its smart contract capabilities, presents a unique opportunity for automating and securing KYC procedures. Martinez and Gomez (2022) analyze the implementation of smart contracts in KYC processes, illustrating how Ethereum can facilitate automated, compliant, and efficient identity verification.

[5] The integration of blockchain technology into KYC processes also raises important regulatory considerations. Patel and Shah (2021) review the regulatory landscape surrounding blockchain-based KYC solutions, emphasizing the need for a balanced approach that safeguards privacy while ensuring compliance with global AML standards.

Chapter 3

SYSTEM ARCHITECTURE AND TECHNOLOGY

3.1 Architecture Diagram

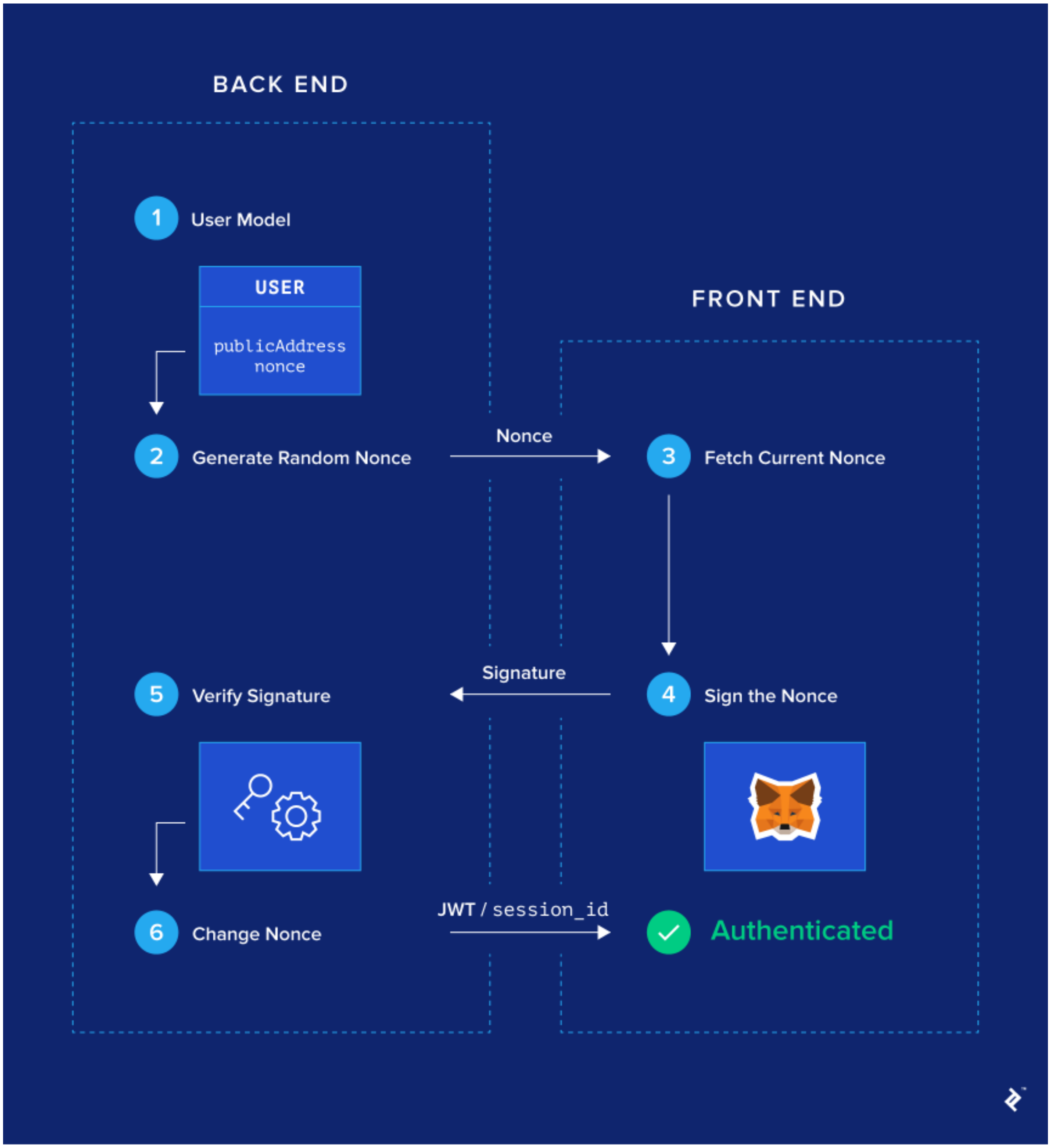


Figure 3.1. Architecture Diagram

1. Login flow initiates by generating one time nonce from the backend and storing the public address to nonce pair in cache.
2. User interacts with Metamask to sign this one time nonce. Metamask uses the user's public address and sings the nonce.
3. The signature generated using metamask is then send to the backend to verify the signature.
4. Over the backend, given the signature from the front-end and the nonce stored in cache in the step 1, the signature is verified using the elliptic curve digital signature algorithm giving back the public address of the user.
5. The required authentication is done using the retrieved public address and a jwt token is being generated.
6. Once the user is authenticated, the nonce for that user is updated for security reasons as it should not be reused to sign the transaction again.

3.2 Technology/Tool Used

1. **HTML**:-
    - Structuring the layout of the web pages for the product identification system.
    - Defining the necessary forms and input fields for user interaction.
    - Displaying essential information related to product details and identification.
2. **CSS:-**
    - Styling and formatting HTML elements to create a visually appealing and consistent user interface
    - Ensuring a responsive design for seamless user experience across various devices.
3. **JavaScript :-**
    - Adding dynamic functionality to enhance user interactions and responsiveness.
    - Implementing real-time updates for product information and transaction history.
    - Managing asynchronous operations for a smoother user experience
4. **Solidity:** Smart Contract Development
    - Writing smart contracts to define rules and logic for product identification on the blockchain.
    - Ensuring the security and integrity of transactions through blockchain-based verification.
    - Creating a tamper-resistant environment by leveraging blockchain technology.
5. **MetaMask** :-Cryptocurrency wallet and browser extension that allows users to interact with decentralized applications (DApps) on the Ethereum blockchain.
6. **Ganache :-**local blockchain emulator that facilitates Ethereum development by providing a simulated blockchain environment for testing and debugging decentralized applications .

Chapter 4

IMPLEMENTATION

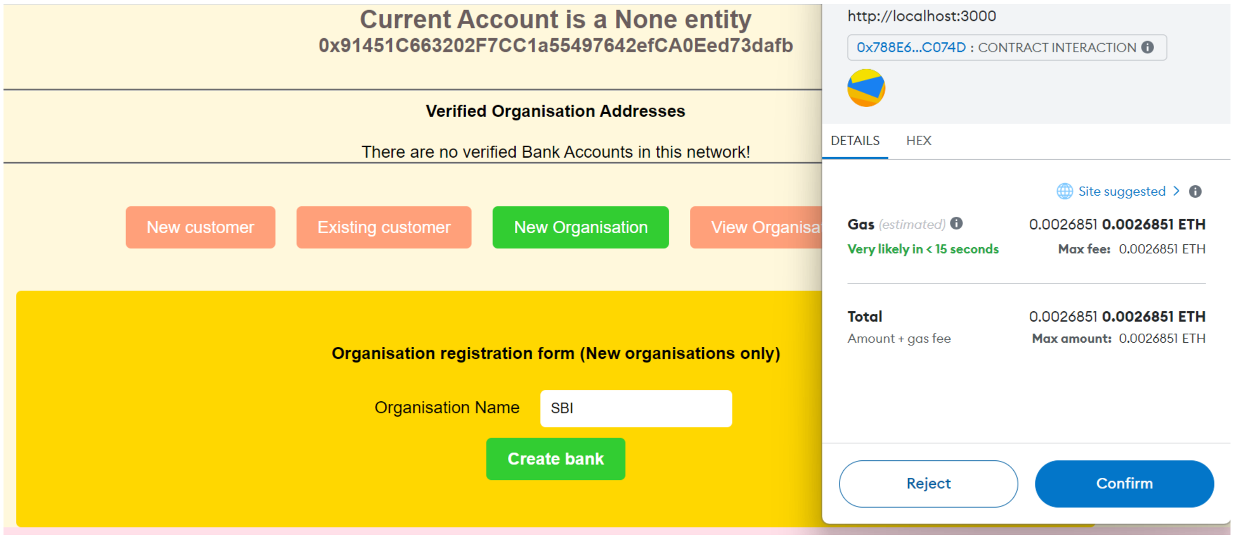
4.1 KycBlochChain.sol

```
pragma solidity ^0.8.12;


contract KycBlockChain is KYC_Functions{

    address[] public Banks;
    address[] public Requests;
    uint public bankslength=0;

    enum Entity { Customer, Organisation }


    struct Customer{
        string c_name;
        string data_hash;
        address bank_address;
        bool exists;
        Entity entity;
    }

    struct Organisation{
        string b_name;
        bool exists;
        Entity entity;
        mapping(address => Status) requests;
        address[] allrequests;
    }

    mapping(address => Customer) allCustomers;
    mapping(address => Organisation) allOrganisations;

    function isOrg() public view returns(bool){
        if(allOrganisations[msg.sender].exists){
            return true;
        }
        return false;
    }
```

```
function isCus() public view returns(bool){
    if(allCustomers[msg.sender].exists){
        return true;
    }
    return false;
}

function newCustomer(string memory _name, string memory _hash, address _bank) public payable
     returns(bool){
    require(!isCus(),"Customer Already Exists!");
    require(allOrganisations[_bank].exists,"No such Bank!");
    allCustomers[msg.sender].c_name = _name;
    allCustomers[msg.sender].data_hash = _hash;
    allCustomers[msg.sender].bank_address = _bank;
    // allCustomers[msg.sender].access[msg.sender] = true;
    allCustomers[msg.sender].exists = true;
    allCustomers[msg.sender].entity = Entity.Customer;
    notifyBank(_bank);
    return true;

}

function newOrganisation(string memory _name) public payable returns(bool){
    require(!isOrg(),"Organisation already exists with the same address!");
    allOrganisations[msg.sender].b_name = _name;
    allOrganisations[msg.sender].exists = true;
    allOrganisations[msg.sender].entity = Entity.Organisation;
    Banks.push(msg.sender);
    bankslength++;
    return true;
}

function viewCustomerData(address _address) public view returns(string memory){
    require(isOrg(),"Access Denied");
    if(allCustomers[_address].exists){
        return allCustomers[_address].data_hash;
    }
    return "No such Customer in the database";
}

function modifyCustomerData(string memory _name,string memory _hash, address _bank) public
     payable returns(bool){
    require(isCus(),"You are not a customer");
    allCustomers[msg.sender].c_name = _name;
    allCustomers[msg.sender].data_hash = _hash;
```

```
        allCustomers[msg.sender].bank_address = _bank;
        return true;
    }

    function notifyBank(address _bankaddress) internal {
        allOrganisations[_bankaddress].requests[msg.sender] = Status.Pending;
        allOrganisations[_bankaddress].allrequests.push(msg.sender);
    }

    function checkStatus() public returns(Status) {
        require(isCus(),"You are not a customer");
        address _presbank = allCustomers[msg.sender].bank_address;
        return allOrganisations[_presbank].requests[msg.sender];
    }

    function changeStatusToAccepted(address _custaddress) public payable{
        require(isOrg(),"You are not permitted to use this function");
        address _bank = allCustomers[_custaddress].bank_address;
        require(_bank == msg.sender,"You dont have access to verify this data");
        allOrganisations[msg.sender].requests[_custaddress] = Status.Accepted;
    }

    function changeStatusToRejected(address _custaddress) public payable{
        require(isOrg(),"You are not permitted to use this function");
        address _bank = allCustomers[_custaddress].bank_address;
        require(_bank == msg.sender,"You dont have access to verify this data");
        allOrganisations[msg.sender].requests[_custaddress] = Status.Rejected;
    }

    function viewRequests() public view returns(address[] memory){
        require(isOrg(),"You are not Permitted");
        return allOrganisations[msg.sender].allrequests;
    }

    function viewName(address _address) public view returns(string memory){
        require(isOrg(),"Not an Organisation");
        return allCustomers[_address].c_name;
    }

}
```

Chapter 5:- RESULTS AND OUTPUTS



Current Account is a None entity
0x91451C663202F7CC1a55497642efCA0Eed73dafb

Verified Organisation Addresses

There are no verified Bank Accounts in this network!

Organisation registration form (New organisations only)

Organisation Name: SBI

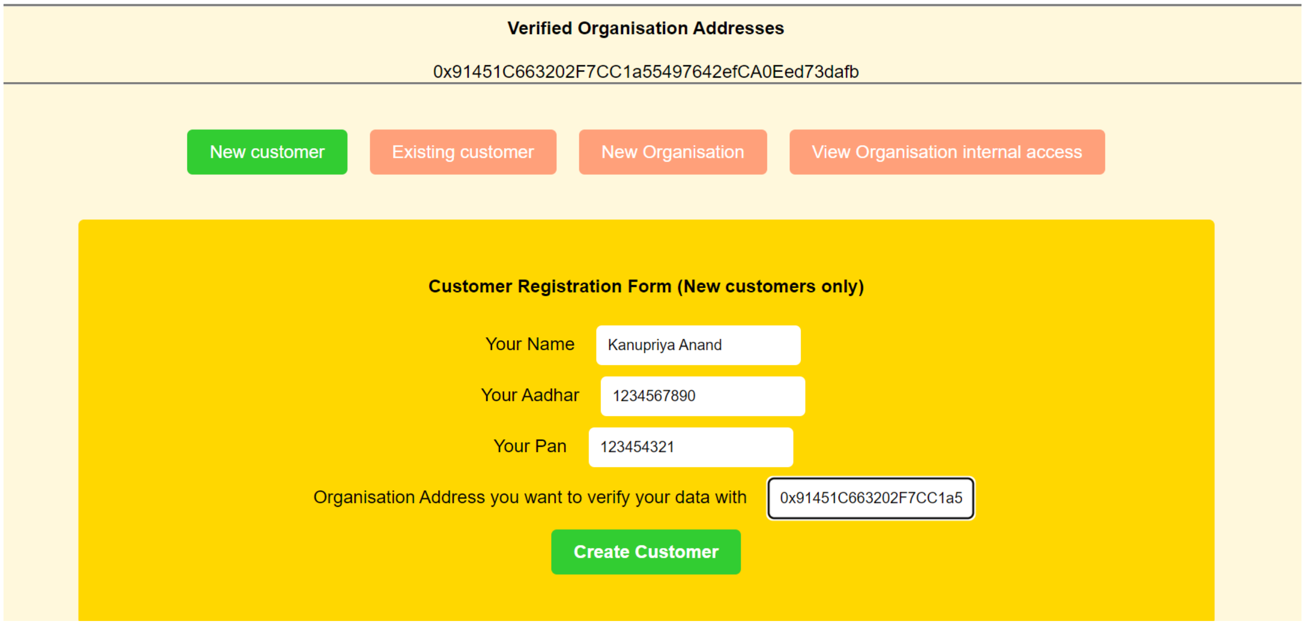
http://localhost:3000

0x788E6...C074D : CONTRACT INTERACTION

Gas (estimated) 0.0026851 0.0026851 ETH
Very likely in < 15 seconds Max fee: 0.0026851 ETH

Total 0.0026851 0.0026851 ETH
Amount + gas fee Max amount: 0.0026851 ETH

Fig 5.1 :- Bank Registration



Verified Organisation Addresses

0x91451C663202F7CC1a55497642efCA0Eed73dafb

Customer Registration Form (New customers only)

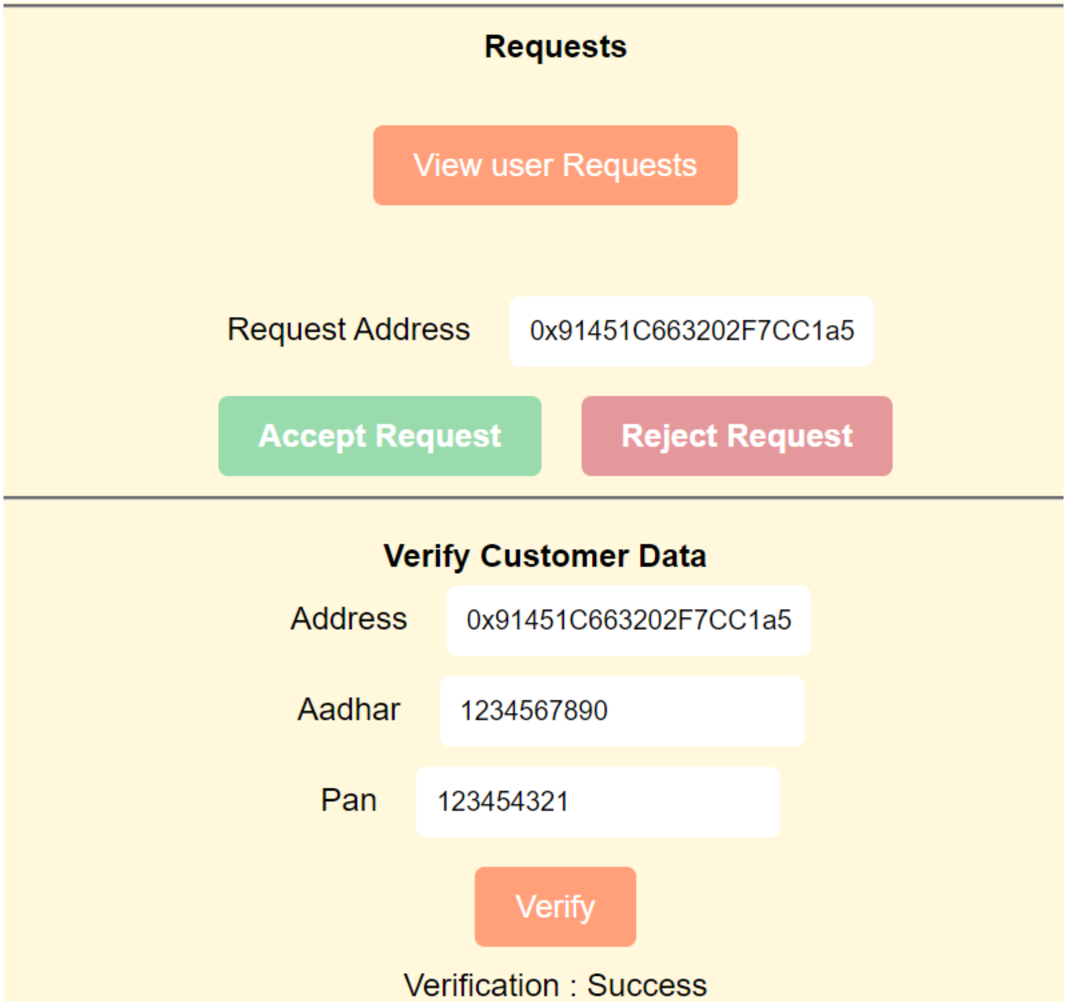
Your Name: Kanupriya Anand

Your Aadhar: 1234567890

Your Pan: 123454321

Organisation Address you want to verify your data with: 0x91451C663202F7CC1a5

Fig 5.2 :- New customer



Requests

Request Address 0x91451C663202F7CC1a5

Verify Customer Data

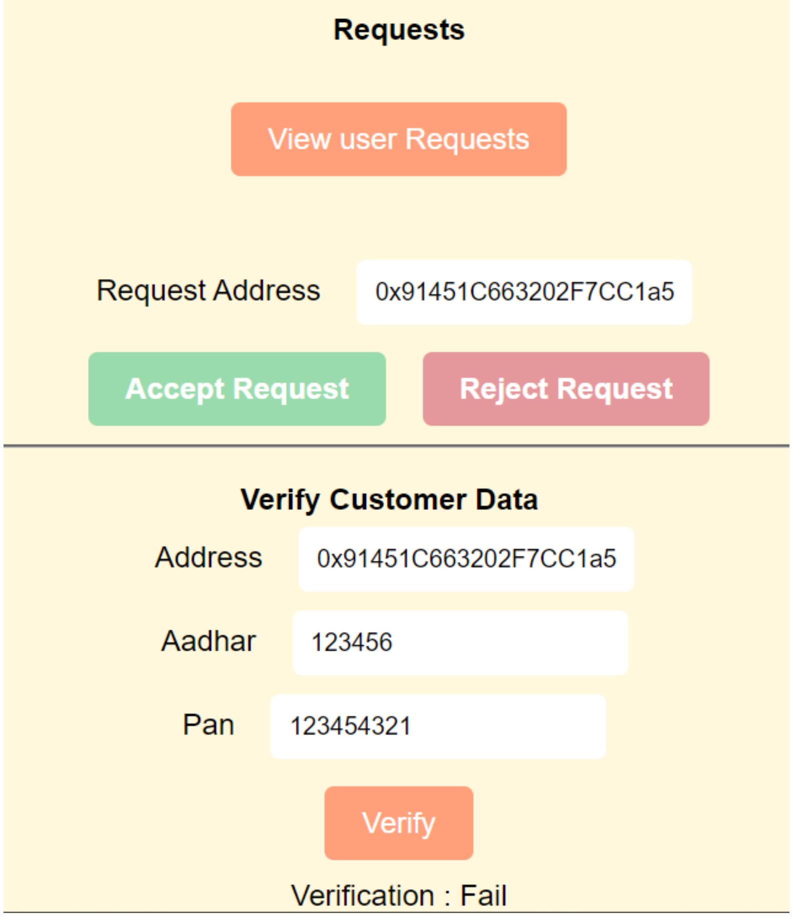
Address 0x91451C663202F7CC1a5

Aadhar 1234567890

Pan 123454321

Verification : Success

Fig 5.3 :- Verified by bank



Request Address: 0x91451C663202F7CC1a5

Verify Customer Data

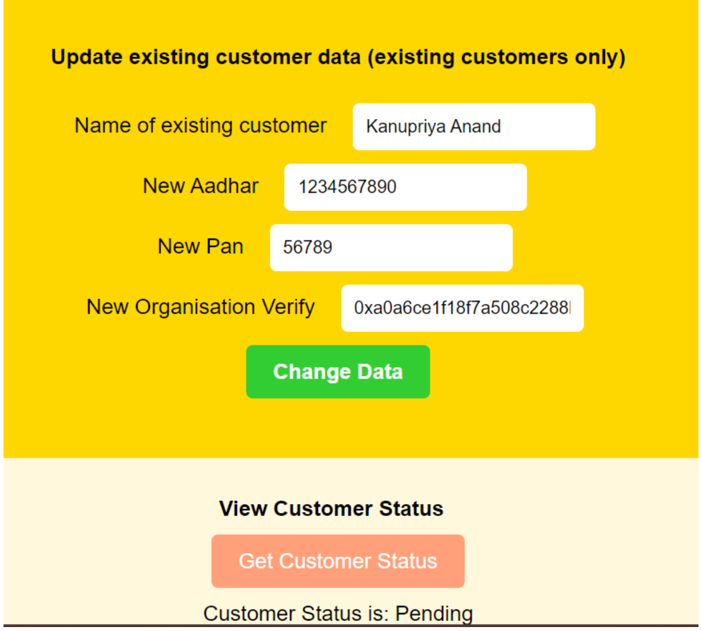
Address: 0x91451C663202F7CC1a5

Aadhar: 123456

Pan: 123454321

Verification : Fail

Fig 5.4 :- Not verified



Update existing customer data (existing customers only)

Name of existing customer: Kanupriya Anand

New Aadhar: 1234567890

New Pan: 56789

New Organisation Verify: 0xa0a6ce1f18f7a508c2288

View Customer Status

Customer Status is: Pending

Fig 5.5 :- Update information

Chapter 6

CONCLUSION AND FUTURE SCOPE

This project has successfully demonstrated the feasibility and benefits of a decentralized Ethereum-based KYC system, addressing the critical challenges faced by traditional KYC processes. Through the development and implementation of this system, we have shown how blockchain technology, specifically Ethereum and its smart contract functionality, can revolutionize the way personal identification and verification processes are conducted in the financial sector.

Key achievements of this project include enhancing data security and privacy for users, streamlining KYC processes to increase efficiency and reduce operational costs, and establishing a framework for standardized and interoperable KYC practices across different institutions and jurisdictions. These advancements represent a significant leap forward in addressing the limitations of centralized KYC systems, such as vulnerability to data breaches, inefficiency, and high costs.

Looking ahead, the decentralized Ethereum-based KYC system stands at the threshold of numerous transformative opportunities. One significant area of future exploration involves the integration with advanced privacy technologies such as zero-knowledge proofs (ZKPs). ZKPs can enable the verification of user identities without disclosing any underlying personal information, thereby elevating the privacy and security dimensions of the KYC process to new heights. This advancement would address one of the most pressing concerns in digital identity verification—balancing the need for compliance and fraud prevention with the imperative to protect individual privacy. Furthermore, extending the system's capabilities to accommodate multi-chain interoperability could vastly enhance its utility and adoption. By enabling the KYC system to operate across various blockchain platforms, it would not only leverage the unique strengths of each platform but also ensure a broader, more resilient infrastructure for identity verification across the global financial ecosystem.

REFERENCES

[1] Smith, J., & Taylor, E. (2020). The evolution of KYC in global finance. Journal of Financial Regulation and Compliance, 28(4), 1-15.

[2] Lee, K. (2019). Blockchain technology and decentralized governance: Is the state still necessary? Journal of Governance and Regulation, 8(1), 8-24.

[3] Anderson, R., Brown, S., & Grant, A. (2021). Blockchain technology in KYC: A comparative study. International Journal of Information Management, 56, 102025.

[4] Martinez, L., & Gomez, J. (2022). Leveraging Ethereum smart contracts for efficient KYC processes. Blockchain in Finance, 3(2), 34-49.

[5] Patel, K., & Shah, D. (2021). Navigating the regulatory maze of blockchain-based KYC processes. Journal of Legal and Financial Studies, 19(3), 45-60.