

Cryptocurrencies and Blockchain Technologies (IT465) Report on

Zero Knowledge and IPFS Based Know Your Customer

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Sachin Prasanna (211IT058)

Abhayjit Singh Gulati (211IT085)

under the guidance of

Prof. Bhawana Rudra



DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

November, 2024

DECLARATION

We hereby *declare* that the Cryptocurrencies and Blockchain Technologies (IT465) Report entitled "***Zero Knowledge and IPFS Based Know Your Customer***", which is being submitted to the **National Institute of Technology Karnataka, Surathkal**, for the award of the Degree of Bachelor of Technology in Information Technology, is a *bonafide report of the work carried out by us*. The material contained in this Cryptocurrencies and Blockchain Technologies Report has not been submitted to any University or Institution for the award of any degree.

Name of the Student (Registration Number) with Signature

- (1) Sachin Prasanna (211IT058)
- (2) Abhayjit Singh Gulati (211IT085)

Department of Information Technology

Place : NITK, Surathkal

Date : 07/11/2024

CERTIFICATE

This is to *certify* that the Cryptocurrencies and Blockchain Technologies (IT465) Report entitled "***Zero Knowledge and IPFS Based Know Your Customer***" submitted by

Name of the Student (Registration Number)

(1) Sachin Prasanna (2110093)

(2) Abhayjit Singh Gulati (2110293)

as the record of the work carried out by them, is *accepted as the B.Tech. Cryptocurrencies and Blockchain Technologies work report submission* in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Technology in the Department of Information Technology, NITK Surathkal.

Signature of Guide with date

Prof. Bhawana Rudra

Assistant Professor (Grade I)

Department of Information Technology

NITK Surathkal-575025

ACKNOWLEDGEMENT

We are profoundly grateful to Prof. Bhawana Rudra for her invaluable guidance and unwavering support. Her expertise and encouragement have been instrumental in shaping the problem statement and the execution of the project until now. Her valuable feedback which we could incorporate in our project during the Mid Semester evaluation. We would also like to extend our sincere thanks to the Information Technology department of National Institute of Technology Karnataka for always providing the resources and an inspiring environment that will be of great help going forward in the project. Finally, we are deeply thankful to our parents for their constant love, patience, and encouragement.

ABSTRACT

The Know Your Customer (KYC) process is essential for institutions to verify client identities, yet traditional methods often incur high costs and pose privacy risks. This paper explores how blockchain technology, with its decentralization and immutability, can enhance the KYC process by improving security and reducing costs. Additionally, it highlights the use of Zero-Knowledge Proofs (ZKPs) as a privacy-preserving solution, allowing identity verification without revealing sensitive information. By integrating ZKPs with blockchain, the paper offers a framework for more secure, efficient, and private KYC systems

Keywords– Interplanetary File System, Zero Knowledge Proofs, Blockchain, Know Your Customer, Solidity, Python

CONTENTS

LIST OF FIGURES	8
1 INTRODUCTION	9
1.1 Overview	9
1.2 Motivation	10
2 LITERATURE REVIEW	11
2.1 Background and Related Works	11
2.2 Problem Statement	12
2.3 Objectives of the Project	13
2.3.1 Data Privacy and Security	13
2.3.2 Decentralized and Immutable Storage	13
2.3.3 Efficient Verifiability	13
3 PROPOSED METHODOLOGY	14
3.1 User Device and Input	15
3.2 IPFS Storage	15
3.3 Blockchain Ledger	15
3.4 Zero-Knowledge Proofs (ZKP) Verification	15
3.5 Administrator and Regulatory Role	16
4 WORK DONE	17
4.1 Smart Contract Overview	17
4.2 Zero Knowledge Proof	17
4.3 Interplanetary File System	18
4.4 Flask Application	19
4.4.1 Application Initialization	19
4.4.2 Helper Functions	19
4.4.3 KYC Registration Route	19
4.4.4 KYC Update Route	20
4.4.5 Admin and Bank Routes	20
4.4.6 Admin and Bank Authentication	20
4.4.7 Blockchain Transactions	20

4.4.8	IPFS Integration	21
4.4.9	Zero-Knowledge Proof Integration	21
5	RESULTS	22
6	CONCLUSIONS AND FUTURE WORK	30
6.1	Conclusions	30
6.2	Future Work	30
	REFERENCES	32

LIST OF FIGURES

3.0.1 Proposed Architecture of the System	14
5.0.1 Main Landing Page	22
5.0.2 KYC Form	23
5.0.3 Successful Registration of KYC	23
5.0.4 Information of the client stored on the IPFS	24
5.0.5 Ganache Local Blockchain	24
5.0.6 To enter Bank Mode	25
5.0.7 Verifying age of Client using Zero Knowledge Proof	25
5.0.8 Admin Mode capabilities	26
5.0.9 Checking if KYC report is valid or expired.	27
5.0.10 Viewing information of a client	27
5.0.11 Count of clients in the Blockchain	28
5.0.12 Error Handling	28
5.0.13 Error Message	29

CHAPTER 1

INTRODUCTION

1.1 Overview

In today’s financial environment, institutions like banks and corporations interact with a broad and varied client base, including both retail and corporate clients. A foundational element of these interactions is the **Know Your Customer (KYC)** process, which plays a critical role in verifying client identities. KYC compliance is a legal mandate, requiring financial entities to meet regulatory standards for both new and existing clients. However, traditional KYC procedures impose a significant financial burden, as institutions must cover rising regulatory costs to remain compliant [1].

Conventional KYC processes typically rely on centralized systems that can compromise privacy, lack efficiency, and incur substantial operational costs. Blockchain technology offers a promising alternative due to its core features of decentralization, immutability, and trustlessness [1]. By leveraging these qualities, blockchain can enhance the integrity and security of KYC processes while reducing the need for intermediaries, thus addressing some of the most pressing challenges in KYC compliance.

Among privacy-preserving techniques, **Zero-Knowledge Proofs (ZKPs)** are particularly suitable for KYC. ZKPs allow one party to verify information’s validity without disclosing the information itself. Compared to other cryptographic approaches like homomorphic encryption or secure multiparty computation, ZKPs offer simpler security assumptions and broader application potential. This makes them a strong candidate for implementing secure, privacy-focused KYC systems on blockchain platforms [2].

Alongside blockchain and ZKP, the InterPlanetary File System (IPFS) adds another essential layer to the process by providing an efficient, decentralized means of storing and sharing large files. IPFS uses cryptographic hashing to uniquely identify and locate files, making it compatible with blockchain technology where file hashes can be stored on-chain, while the data itself remains off-chain. This hybrid approach

offers an efficient way to handle high data volumes without overloading blockchain storage [3].

1.2 Motivation

The integration of blockchain and privacy-preserving technologies like ZKPs into KYC processes is motivated by the need for a secure, efficient, and privacy-conscious identity verification system. Traditional KYC approaches face significant limitations, including high costs, human error, and dependency on centralized authorities, which can lead to privacy risks and inefficiencies. The decentralized nature of blockchain not only addresses these issues by enabling transparent, tamper-resistant data storage but also allows institutions to reduce redundant identity verification efforts, a significant pain point in KYC practices [4].

By using ZKPs, financial institutions can verify essential customer information without exposing sensitive data. This privacy-preserving capability is invaluable in KYC, where confidentiality and security are paramount. Additionally, integrating the InterPlanetary File System (IPFS) offers a scalable solution for managing and sharing large files, as it provides a decentralized storage method using cryptographic hashes, which can then be anchored on a blockchain [3]. Together, ZKP, blockchain, and IPFS create a cohesive solution aimed at transforming the KYC process into a more secure, efficient, and cost-effective system, opening new avenues for privacy-preserving verification within financial services.

CHAPTER 2

LITERATURE REVIEW

2.1 Background and Related Works

Blockchain is a decentralized, continuously growing ledger of records validated by network participants. Initially developed for managing cryptocurrency transactions, blockchain’s distributed ledger technology has expanded into various domains, including healthcare [5]. Its decentralized architecture relies on peer-to-peer (P2P) networks, cryptographic techniques, and consensus algorithms to ensure secure and transparent data storage without the need for central authorities [6].

The key advantages of blockchain technology—decentralization, immutability, and traceability—make it ideal for systems requiring high security and integrity. These features have led to its use in building tamper-resistant data storage solutions and in designing privacy-preserving transactions, enabling secure interactions between parties without revealing sensitive information [7].

In the process of enrolling new customers or updating existing ones, KYC has become a crucial requirement for institutions such as banks and insurance companies to verify their clients’ involvement in any illegal activities [8]. However, the current KYC process faces several challenges that create inefficiencies for both customers and organizations. These include Human Error, Lack of Skilled Personnel, Malicious Users, Duplication and Costs, Time Delays, Reputational and Regulatory Risks, Lack of International Standards [9]

To address the challenges of traditional KYC processes, the concept of utilizing blockchain technology for KYC has emerged. Malhotra et al. [4] propose a solution where financial institutions implement KYC using a private blockchain. In this approach, a client’s identity is verified only once, and the verified information is securely stored on the blockchain. This allows other financial institutions to access the stored data for future verification, eliminating the need for repeated KYC checks.

In the approach given by Yadav et al.[1] after the details provided by the user are authenticated, all the data provided by the user in the KYC form is then added

into the Blockchain. They use the Ethereum API for building the Blockchain using solidity language to create a smart contract. The user is then notified that his/her details have been verified and he/she can now proceed to apply to banks where he/she wishes to open an account. The user will be presented a list of banks wherein which he/she simply has to tap on the list item of that bank.

To further enhance KYC security, Zero-Knowledge Proofs (ZKP) can be utilized. ZKPs allow users to prove possession of certain information without disclosing the information itself, offering an additional layer of privacy [10]. The application of ZKPs on the blockchain has gained popularity recently. For example, Partala et al. review various ZKP schemes applied to confidential transactions and private smart contracts on blockchains [11]. Li et al. propose a decentralized, privacy-preserving architecture that integrates ZKPs into blockchain-based traffic management systems to ensure data integrity and privacy [12].

In addition, we will leverage InterPlanetary File System (IPFS) to efficiently store and share large files. IPFS uses cryptographic hashes, which can be easily stored on a blockchain. [3].

2.2 Problem Statement

In a traditional KYC (Know Your Customer) process, users need to provide sensitive personal information such as name, address, and date of birth to service providers like banks, insurance companies, or government agencies. This data is typically stored in centralized databases, raising concerns about privacy, data breaches, and unauthorized access. Furthermore, service providers often require specific information (e.g., verifying that a person is over 18 or resides in a certain country), but not the full details of the person’s identity or address. The challenge is to create a system that allows users to prove certain facts about themselves (e.g., age, residence) without revealing sensitive information.

2.3 Objectives of the Project

2.3.1 Data Privacy and Security

Implementing a decentralized KYC system using Zero- Knowledge Proofs (ZKPs) to ensure that users can prove their attributes (e.g., age, residency) without revealing their full personal details. Sensitive data is encrypted and stored on the InterPlanetary File System (IPFS), ensuring privacy and protection from unauthorized access.

2.3.2 Decentralized and Immutable Storage

Utilizing blockchain technology to record cryptographic proofs of KYC updates and changes, ensuring that no centralized authority can alter the records. This creates a tamper-proof, auditable ledger for KYC compliance, enabling transparency and accountability.

2.3.3 Efficient Verifiability

Enabling service providers to verify specific claims about a user's identity or attributes through Zero-Knowledge Proofs, without accessing the full KYC data. The blockchain provides a secure and automated method for verification, significantly reducing manual checks and maintaining privacy.

CHAPTER 3

PROPOSED METHODOLOGY

Figure 3.0.1 illustrates the architecture of our proposed de- centralized KYC system, integrating Zero-Knowledge Proofs (ZKPs), IPFS, and blockchain. The architecture ensures se- cure, private, and verifiable KYC data management.

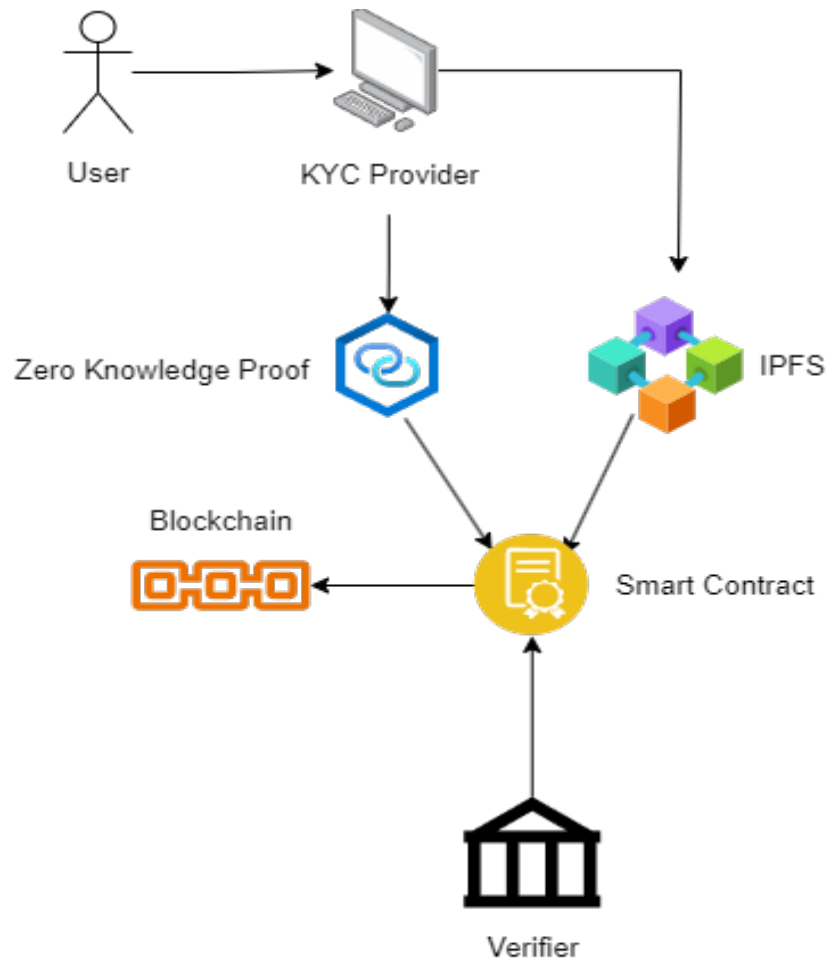


Figure 3.0.1: Proposed Architecture of the System

3.1 User Device and Input

The user initiates the KYC process through their device, entering personal data such as name, date of birth, address, and other identity-related information.

3.2 IPFS Storage

The encrypted KYC data is stored in the InterPlanetary File System (IPFS). The IPFS system provides decentralized storage, ensuring that the data remains accessible even in the absence of a centralized server. The system generates a unique content identifier (CID) for the stored data, which can be used to retrieve the data when needed.

3.3 Blockchain Ledger

The CID generated from IPFS is recorded on the blockchain. This immutable ledger ensures that once the KYC data is uploaded and verified, the proof of the KYC and any changes made to it are permanently stored, making the system tamper-proof and auditable.

3.4 Zero-Knowledge Proofs (ZKP) Verification

Service providers can request specific information from the user without accessing the full KYC data. This speeds up the process greatly. Using ZKPs, the user can prove these attributes without revealing the underlying sensitive information. The blockchain is leveraged to validate these ZKPs, ensuring that the verification process is secure and efficient. We are using ZKPs for verifying if the age of user is above 18 or not.

3.5 Administrator and Regulatory Role

An administrative authority monitors and regulate KYC processes. Their role includes overseeing policy compliance and ensuring that the system adheres to relevant regulatory requirements.

CHAPTER 4

WORK DONE

4.1 Smart Contract Overview

The KYC smart contract provides a secure and automated way to handle client information for KYC verification. This contract enables the registration of clients, storing essential details such as a report URI, expiration date of the KYC verification, and a zero-knowledge proof (ZKP) used for age verification. By using this contract, an organization can ensure that each client's information remains up-to-date, accessible, and valid over time.

The contract includes functions to register new clients, update existing client information, and check the validity of each client's KYC report based on an expiration date. Only the contract administrator has the authority to monitor and flag KYC reports that are nearing expiration, which helps in timely updating and compliance management. The contract also uses event logs to notify about new client registrations, information changes, and expiring KYC reports, allowing transparent tracking of these updates on the blockchain.

Through this setup, the KYC smart contract offers an efficient, reliable, and transparent way to manage client identity verification and compliance with minimal manual intervention.

4.2 Zero Knowledge Proof

The zero-knowledge proof mechanism in this system allows for age verification without directly revealing a client's age, ensuring privacy. When a client's age is evaluated, it is transformed into a simple identifier based on whether the client is above or below a specified threshold, such as 18 years. This identifier is then used to produce a unique 32-character string, or "proof," that serves as a representation of the client's age eligibility.

To create this proof, each character in the identifier contributes to a calculated

hash value. This value undergoes further transformations to produce a consistent and pseudo-random sequence of 32 characters composed of letters and digits. This hash generation process ensures that even with the same age category, the output remains difficult to reverse-engineer, effectively preserving the client's anonymity.

In this way, the zero-knowledge proof allows the system to verify that a client meets the age requirement without revealing any additional information about the client's specific age or other personal details. This approach balances the need for verification with the need for data privacy, creating a secure and reliable method of proof.

4.3 Interplanetary File System

The Interplanetary File System (IPFS) is utilized in this system to securely store and retrieve client KYC data in a decentralized manner. IPFS provides a distributed network for storing files, where each file is identified by a unique hash, ensuring data integrity and accessibility. This approach aligns well with blockchain applications, where decentralization and security are paramount.

In this implementation, client information such as name, date of birth, age, email, nationality, occupation, and zero-knowledge proof (ZKP) is structured in JSON format. This JSON data is then submitted to the IPFS network through the Pinata API, a platform that facilitates file pinning on IPFS. Once the JSON is pinned to IPFS, a unique hash (known as the IPFS CID) is returned. This hash serves as a secure reference to the data and can be used to retrieve the information from IPFS when needed.

By leveraging IPFS, this system ensures that sensitive KYC data is stored in a decentralized environment, enhancing data resilience and privacy. The integration of IPFS with blockchain-based KYC processes allows for a more secure and efficient management of client information.

4.4 Flask Application

The Flask application serves as the primary interface for handling KYC registration, updates, and verification through an Ethereum-based smart contract. The application integrates various components, including IPFS for decentralized storage, a smart contract for KYC data management, and zero-knowledge proof (ZKP) for privacy-preserving age verification.

4.4.1 Application Initialization

The application begins by importing necessary modules, including `Flask`, `render_template`, `request`, and `flash` for web functionalities, as well as `Web3` for blockchain interactions. After initializing the `Flask` app, the KYC smart contract is loaded through the `initContract()` function, allowing the app to interact with the Ethereum blockchain.

4.4.2 Helper Functions

- **calculate_age(dob):** This helper function calculates a user's age based on the date of birth (DOB) provided in `dd/mm/yyyy` format. It calculates the age by comparing the birth date with the current date.

4.4.3 KYC Registration Route

The KYC registration route (`/register`) handles POST requests and collects user data from a form submission. The steps involved include:

1. **Data Collection:** The form collects personal details, including first and last names, date of birth, email, nationality, occupation, and user ID.
2. **ZKP Proof Generation:** The application uses a zero-knowledge proof to verify the user's age. The ZKP is generated based on the calculated age to allow age verification without exposing exact details.
3. **Data Conversion and Storage in IPFS:** The collected data is structured into JSON format and uploaded to IPFS through Pinata, resulting in a unique IPFS URI.

4. **Smart Contract Interaction:** The KYC smart contract is invoked to register the user with their KYC data and ZKP proof. The app then waits for a transaction receipt and provides a confirmation message if successful.

4.4.4 KYC Update Route

The KYC update route (`/update`) allows users to update their KYC data. Similar to registration, it collects personal data, generates a new ZKP, uploads the data to IPFS, and updates the KYC contract with the new IPFS URI. This route is useful for users who need to modify their information after registration.

4.4.5 Admin and Bank Routes

The application provides distinct access for administrators and banking officials to perform the following operations:

- **Admin Page** (`/admin_page`): The admin can verify the validity of KYC records, retrieve client counts, and access client information.
- **Bank Page** (`/bank_page`): The bank page allows officials to verify the user's age based on the ZKP, which confirms eligibility for age-restricted services.

4.4.6 Admin and Bank Authentication

The `/admin_login` route manages role-based access control, requiring specific passwords for administrators and bank personnel. On successful login, administrators and bank officials are directed to their respective pages.

4.4.7 Blockchain Transactions

To interact with the Ethereum blockchain, the app utilizes Web3 for sending transactions to the KYC smart contract. Functions include:

- **registerKYC:** Registers new KYC data on the blockchain with an IPFS URI and ZKP.
- **updateKYC:** Updates an existing KYC record with new data.

- **checkValidity**: Checks if a user's KYC is valid.
- **getClientCount**: Retrieves the total number of registered KYC clients.
- **Clientdatabase**: Provides detailed information about a specific client.

4.4.8 IPFS Integration

Data is securely stored on IPFS through Pinata, ensuring decentralization and data integrity. The IPFS URI generated is stored in the smart contract, enabling efficient data retrieval without directly storing sensitive data on the blockchain.

4.4.9 Zero-Knowledge Proof Integration

The application leverages zero-knowledge proofs to allow users to prove age eligibility without revealing their exact age. The ZKP is generated based on the user's age, allowing verification without exposing sensitive data.

CHAPTER 5

RESULTS

This section presents the results of our project, with each figure illustrating a key part of the system and functionality. Descriptions are provided to clarify the user experience at each stage.

The user initially lands on the home page, where they can navigate to the desired section based on their role and preferences. Figure 5.0.1 displays the main landing page.

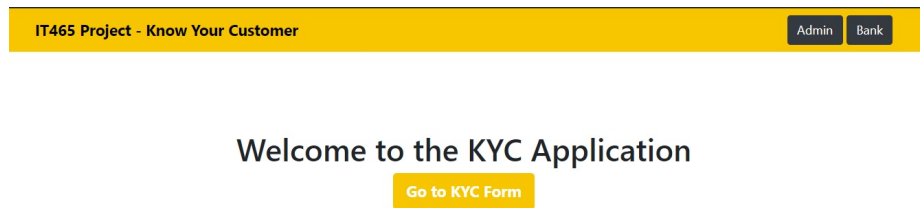


Figure 5.0.1: Main Landing Page

To submit a KYC application, the user can select the **Go to KYC Form** button, which opens a form to enter their personal information. Figure 5.0.2 shows the KYC form interface.

Once the user successfully registers for the KYC, A Upon successful registration, a prompt appears, confirming the KYC registration and providing the Zero Knowledge Proof (ZKP), IPFS URI, and transaction receipt. This process is shown in Figure 5.0.3.

IT465 Project - Know Your CustomerAdminBank

Fill in your Data (KYC):

First Name:
sachin

Last Name:
prasanna

Date of Birth (dd/mm/yyyy):
23/07/2002

Email:
sachin@gmail.com

Nationality:
indian

Occupation:
student

User ID:
0xd7d43471045d4aC7F569f76e977BeEA85052583a

Register

Update

Figure 5.0.2: KYC Form

IT465 Project - Know Your CustomerBack

KYC Registration Successful

User ID:
0xd7d43471045d4aC7F569f76e977BeEA85052583a

ZKP Proof:
012ef85c20ae5a6f849a85e193a513e32b9a1bbbdcca89aa6cde414b51f7938b

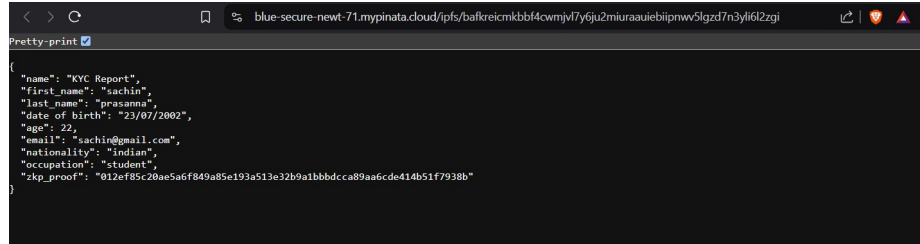
Report IPFS URI:
ipfs://bafkreicmkbbf4cwmjvl7y6ju2miuraauiabiipnwv5lgzd7n3yli6l2zgi

Transaction Receipt:
AttributeDict({'transactionHash': HexBytes('0x52b82403186de0e6753447752a726a0f3daf4ea13921f

Go Back

Figure 5.0.3: Successful Registration of KYC

After registration of data, the record is stored on the blockchain and the information is pinned to the IPFS via Pinata. This is shown in Figure 5.0.4



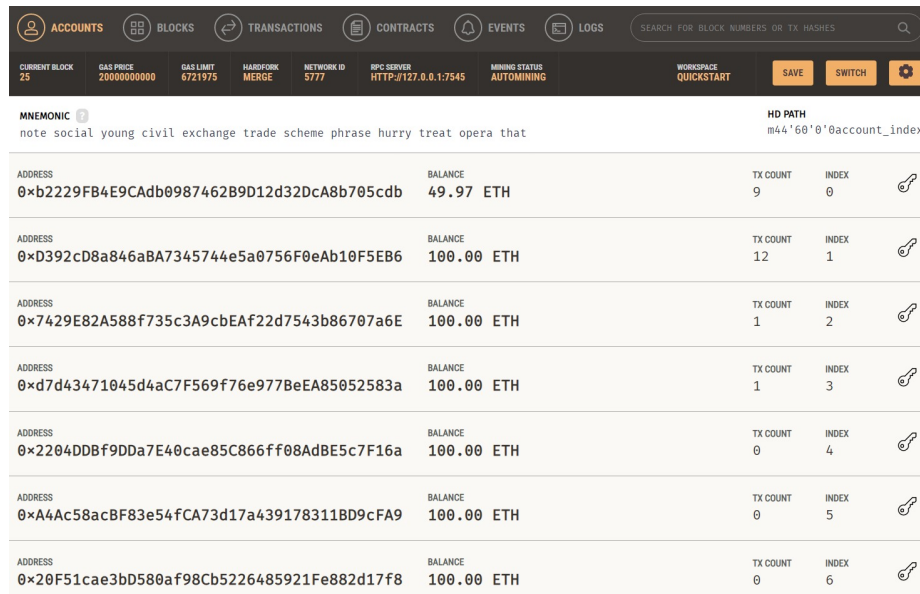
```

{
  "name": "KYC Report",
  "first_name": "sachin",
  "last_name": "prasanna",
  "date of birth": "23/07/2002",
  "age": 22,
  "email": "sachin@gmail.com",
  "nationality": "indian",
  "occupation": "student",
  "zkp_proof": "012ef85c20ae5a6f849a85e193a513e32b9a1bbddca89aa6cde414b51f7938b"
}

```

Figure 5.0.4: Information of the client stored on the IPFS

Figure 5.0.5 gives us a glimpse of the Ganache UI and how many accounts are present in the chain. They will be used for creation of the creation of the KYC process.



ADDRESS	BALANCE	TX COUNT	INDEX
0xb2229FB4E9CAdb0987462B9D12d32DcA8b705cdb	49.97 ETH	9	0
0xD392cD8a846aBA7345744e5a0756F0eAb10F5EB6	100.00 ETH	12	1
0x7429E82A588f735c3A9cbEaf22d7543b86707a6E	100.00 ETH	1	2
0xd7d43471045d4aC7F569f76e977BeEA85052583a	100.00 ETH	1	3
0x2204DDBf9DDa7E40cae85C866ff08AdBE5c7F16a	100.00 ETH	0	4
0xA4Ac58acBF83e54fCA73d17a439178311BD9cFA9	100.00 ETH	0	5
0x20F51cae3bd580af98Cb5226485921Fe882d17f8	100.00 ETH	0	6

Figure 5.0.5: Ganache Local Blockchain

The User Interface also allows a user to enter into the Bank Mode. To do so he/she must enter the password as shown in Figure 5.0.6

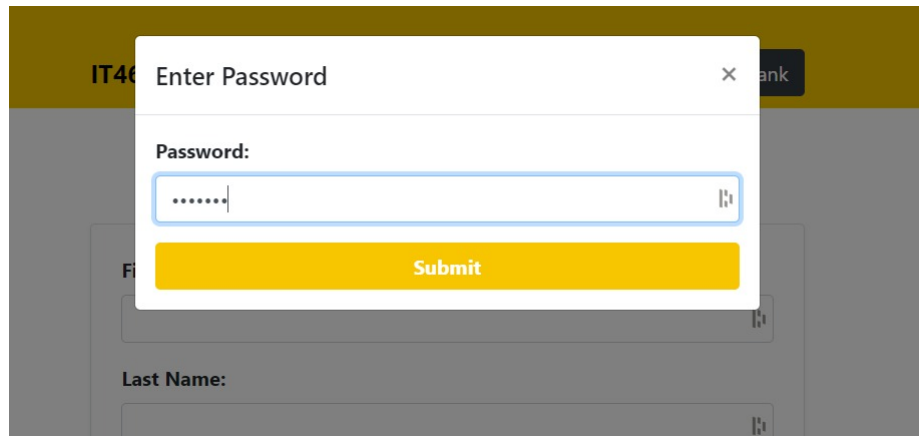
A screenshot of a web application interface. A modal dialog box titled "Enter Password" is centered on the screen. It has a close button (X) in the top right corner. Inside the dialog, there is a label "Password:" followed by a text input field containing seven dots. To the right of the input field is a small icon. Below the input field is a yellow button labeled "Submit". In the background, a yellow header bar contains the text "IT465 Project - Know Your Customer" and a "Back" button. Below the header, there is a "Verify Age" section with a "User ID:" label and a text input field containing a long alphanumeric string. A yellow button labeled "Verify Age" is below this field. At the bottom, there is a message "Verified: User is 18 or older".

Figure 5.0.6: To enter Bank Mode

In Bank Mode, bank personnel can verify a client's age using the Zero Knowledge Proof. Figure 5.0.7 demonstrates this process.

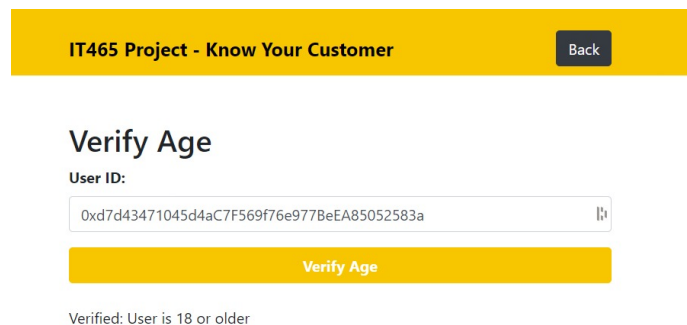
A screenshot of a web application interface. At the top, a yellow header bar contains the text "IT465 Project - Know Your Customer" and a "Back" button. Below the header, there is a "Verify Age" section. It starts with the title "Verify Age" in bold. Below the title is a label "User ID:" followed by a text input field containing a long alphanumeric string. To the right of the input field is a small icon. Below the input field is a yellow button labeled "Verify Age". At the bottom of the section, there is a message "Verified: User is 18 or older".

Figure 5.0.7: Verifying age of Client using Zero Knowledge Proof

One can also enter the admin mode and utilise all the features available to the admin. The admin mode can be as seen in Figure 5.0.8.

The screenshot displays the 'Admin Dashboard' interface. At the top, a yellow header bar contains the text 'IT465 Project - Know Your Customer' on the left and a 'Back' button on the right. Below the header, the main content area is divided into three sections. The first section, 'Admin Dashboard', has a sub-header 'Check KYC Validity'. It includes a label 'Enter User ID:' followed by a text input field containing the hexadecimal string '0xd7d43471045d4aC7F569f76e977BeEA85052583a'. A yellow 'Check' button is positioned below the input field. The second section is titled 'Get Client Count' and features a single yellow button labeled 'Get Client Count'. The third section, 'Client Database Information', also starts with the label 'Enter User ID:' and a text input field, which is currently empty. A yellow 'Get Info' button is located below this input field.

Figure 5.0.8: Admin Mode capabilities

The admin has the capability to check whether the the KYC report is valid or not. This can be seen as in Figure 5.0.9

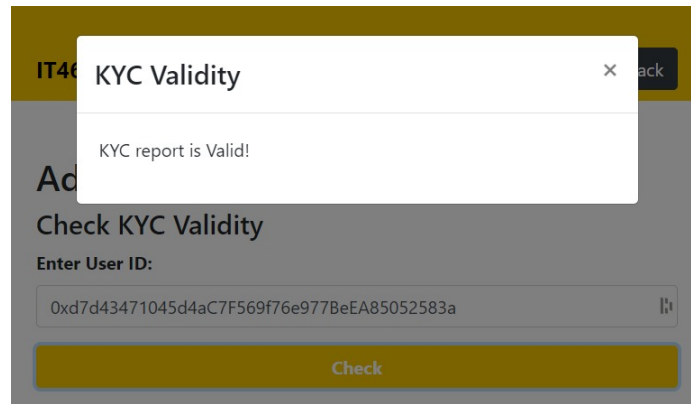


Figure 5.0.9: Checking if KYC report is valid or expired.

Admins can also retrieve client information stored on the blockchain using a user-name, as depicted in Figure 5.0.10.

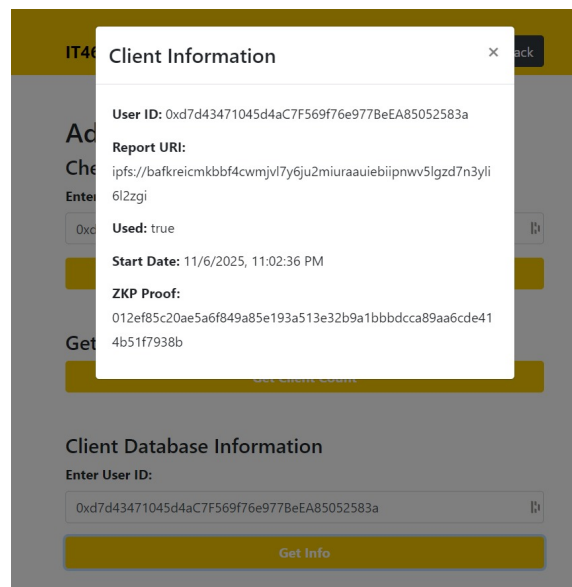


Figure 5.0.10: Viewing information of a client

The system provides a count of all registered clients, viewable in Figure 5.0.11.

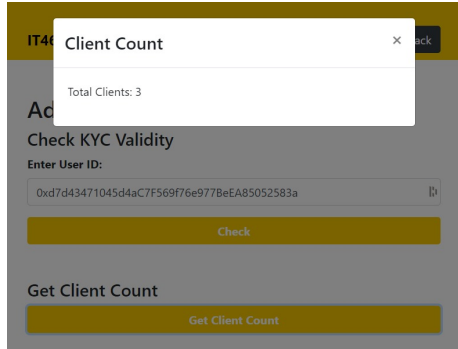


Figure 5.0.11: Count of clients in the Blockchain

If a user attempts to register with a previously used ID, the system provides an error message, ensuring ID uniqueness. This is shown in Figure 5.0.12.

A screenshot of a web application interface for KYC registration. The header is 'IT465 Project - Know Your Customer' with 'Admin' and 'Bank' buttons. Below the header is a section titled 'Fill in your Data (KYC):'. The form contains several input fields: 'First Name' (abhayjit), 'Last Name' (singh), 'Date of Birth (dd/mm/yyyy)' (10/10/2002), 'Email' (abhayjit@gmail.com), 'Nationality' (indian), 'Occupation' (student), and 'User ID' (0xd7d43471045d4aC7F569f76e977BeEA85052583a). At the bottom of the form are two buttons: 'Register' and 'Update'.

Figure 5.0.12: Error Handling

Additional error messages, such as when other invalid actions are performed, are shown in Figure 5.0.13.

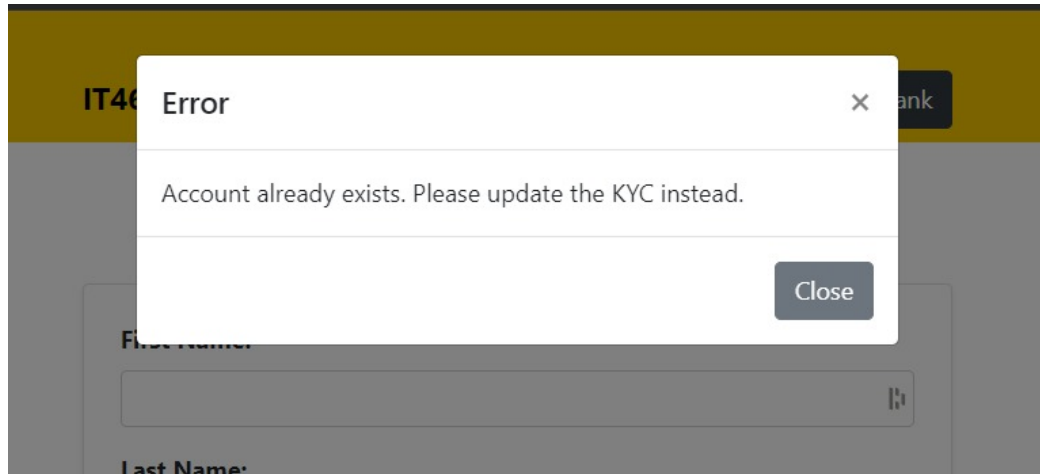


Figure 5.0.13: Error Message

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The project went smoothly with implementing a smart contract for the KYC contract, which enables several functions and including the verification of the Zero Knowledge Proof (ZKP) for verifying if a user is above 18 years of age.

A Flask based python app was built to get in data from the users for the KYC and stored in the IPFS via Pinata. Only information like the Zero Knowledge Proof, IPFS hash were stored on chain, which makes the chain light weight and secure as well.

There is an admin mode functionality which allows the admin to view users information and their KYC validity. The number of clients currently registered can also be seen.

The bank mode allows a bank to verify the whether a user is above 18 years of age or not by using their user id and the Zero Knowledge Proof stored on chain. Banks need to verify their identity with a password before requesting verification of a user's age.

6.2 Future Work

There are several avenues for future enhancement of the KYC application.

Improving the user interface (UI) is also a critical next step. While the current UI is functional, future work could focus on creating a more intuitive and accessible experience for voters of all demographics, including non-tech-savvy individuals and mobile users. This could involve refining mobile support and designing a streamlined, easy-to-navigate interface.

Also, due to the limitation of the Pinata API IPFS system, the KYC information could not be encrypted before pinning it on the IPFS. A future work would be to

encrypt the data before storing it in the IPFS and decrypting it whenever needed again.

More functions can be added in the smart contract for the admin mode as well, which can extend the scope of the application.

Finally, the application should be updated to align with evolving legal and regulatory frameworks.

REFERENCES

- [1] Piyush Yadav and Raj Chandak. Transforming the know your customer (kyc) process using blockchain. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–5, 2019.
- [2] Ryan Lavin, Xuekai Liu, Hardhik Mohanty, Logan Norman, Giovanni Zaarour, and Bhaskar Krishnamachari. A survey on the applications of zero-knowledge proofs. *arXiv preprint arXiv:2408.00243*, 2024.
- [3] Mathis Steichen, Beltran Fiz, Robert Norvill, Wazen Shbair, and Radu State. Blockchain-based, decentralized access control for ipfs. 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)*, pages 1499–1506, 2018.
- [4] Diksha Malhotra, Poonam Saini, and Awadhesh Kumar Singh. How blockchain can automate kyc: Systematic review. *Wireless Personal Communications*, 122(2):1987–2021, 2022.
- [5] Anuraag A Vazirani, Odhran O’Donoghue, David Brindley, and Edward Meinert. Blockchain vehicles for efficient medical record management. *NPJ digital medicine*, 3(1):1, 2020.
- [6] Rahul Johari, Vivek Kumar, Kalpana Gupta, and Deo Prakash Vidyarthi. Blossom: Blockchain technology for security of medical records. *ICT Express*, 8(1):56–60, 2022.
- [7] Shunli Ma, Yi Deng, Debiao He, Jiang Zhang, and Xiang Xie. An efficient nizk scheme for privacy-preserving transactions over account-model blockchain. *IEEE Transactions on Dependable and Secure Computing*, 18(2):641–651, 2020.
- [8] Prakash Chandra Mondal, Rupam Deb, and Mohammad Nurul Huda. Transaction authorization from know your customer (kyc) information in online banking. In *2016 9th international conference on electrical and computer engineering (ICECE)*, pages 523–526. IEEE, 2016.

- [9] Wazen M Shbair, Mathis Steichen, Jérôme François, et al. Blockchain orchestration and experimentation framework: A case study of kyc. In *IEEE/IFIP Man2Block 2018-IEEE/IFIP Network Operations and Management Symposium*, 2018.
- [10] Po-Wen Chi, Yun-Hsiu Lu, and Albert Guan. A privacy-preserving zero-knowledge proof for blockchain. *IEEE Access*, 11:85108–85117, 2023.
- [11] Juha Partala, Tri Hong Nguyen, and Susanna Pirttikangas. Non-interactive zero-knowledge for blockchain: A survey. *IEEE Access*, 8:227945–227961, 2020.
- [12] Wanxin Li, Hao Guo, Mark Nejad, and Chien-Chung Shen. Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach. *IEEE access*, 8:181733–181743, 2020.