Contents

1 Introduction

Marksheets contain information confidential to the individuals and should not be easily accessible to others.Verification of mark sheets is a tedious process for the industries to carry out, which often involves contacting the respective universities/colleges to cross verify the mark sheets. Any particular industry needs time to authenticate if the certificates of any student are genuine.First, the mark sheets of students will be created by university. The higher authorities will authenticate the digital certificate and will mark it with a digital signature and it will be given to the student and its cryptographic hash will be stored on the blockchain and documents stored on IPFS.We will use a smart contract at the backend to interact with the blockchain and the encrypted hash value of each document will be stored in the IPFS which will be verified against the user document.

1.1 Motivation

Every year, millions of students graduate from universities, some of them will continue to pursue higher studies or start job hunting. For admission, students need to produce these certificates in institutions or companies during interviews. Tracking these certificates and validating their authenticity manually becomes a tedious job. As information technology has developed rapidly over the years, data security has become inevitable. Graduation certification can be easily altered these days and sometimes, students lose their educational certificates, and reapplying for hard copies is time consuming, in order to avoid the certificates from getting forged and losing one's certificate, a system using blockchain technology. Blockchain is a decentralized system where all the participating parties are peers and all the transactions take place without third party intermediaries. Every node in the blockchain network must agree when a transaction is made else the transaction would not be considered. Thus, transactions using blockchain technology become more secure.

1.2 Literature survey

The primary goals of the blockchain technology are to provide decentralization, immutability, shared and distributed ledger, provide better data security and faster settlement of transactions.

Higher Education's Certificates Model based on Blockchain Technology(2021) With the advancement of the digital world, the creation of distributed and secure electronic systems has become a hot topic. The use of blockchain technology is a great example of these systems with interesting features like transparency, trust, and decentralisation and centralised data storage with unaltered and permanent recordings The implementation of such. It is vital to incorporate such technology into systems for granting student credentials, especially when the persons are subjected to multiple academic studies and training from various institutions and acquire several certificates. It is critical to preserve these certificates in a form that is permanent, not tampered with, and not counterfeit.The proposed model satisfies the requirements for preventing forgery certificates and managing digital academic credentials, particularly in terms of trust, high outputs, availability, transparency, low costs, and resource consumption, especially given the large number of educational institutions that exist today. The technicalities of implementation are not covered by this study. It is advised that more research be conducted in order to create and assist the construction and implementation of a blockchain solution in education.[1].

Blockchain and Smart Contract for Digital Certificate (IEEE 2018)

The authors of this paper have proposed a digital certificate system based on blockchain technology to reduce the likelihood of graduation certificate forging by providing data security and accuracy. They have developed a decentralized application using smart contracts and designed a certificate system based on Ethereum blockchain. In this system, there are 3 types of users involved, Certification units who grant the certificates and have access to the system, students who are granted certificates after they fulfil certain requirements can access the certificates. And the service provider is responsible for system maintenance. The drawback of their method was that they were using 'one hash as a 'key', which makes it publicly accessible[2].

Certificate Verification using Blockchain and Generation of Transcript (IJERT 2017)

The authors of this paper have designed a system which automatically generates the certificates and also reduces the manual work needed for the verification of the documents. They have developed a decentralized application based on Ethereum and have written the smart contracts using solidity language. There system consists of 3 actors. The college acts as the certificate issuing authority, student can view and download the certificates and is also provided with a unique hash value to access and verify their certificates in the future. The company would be any organization who wants to verify the authentication, originality of the certificate. The original documents are stored in the IPFS and the hash value of every document is stored in the blockchain which helps in preserving the data[3].

An innovative IPFS-Based Storage Model for Blockchain (IEEE 2018)

The authors of this paper have proposed an IPFS-based blockchain data storage model for storing data to overcome the limitations of data storage on blockchain. This is a peer-to-peer model for sharing and storing data on a distributed file system. This model uses a content-addressing to uniquely identify each file in a global namespace connecting all computing devices. Whenever a document is uploaded on IPFS, it is divided into chunks and distributed with the nodes present in the IPFS and the path to access each chunk of the document would be available only with the user.Thus this model can be used for uploading documents, reading documents, and Downloading documents with better storage space and security[4].

An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends (IEEE 2017)

In this paper the authors have given an in-depth understanding of blockchain technology. Many terms used in blockchain technology have been discussed and one of the critical concepts called as Smart contract is also explained in depth.Whenever a block is to be added to the blockchain the block's hash value will be stored in the previous block and a chain of blocks also called as ledger is formed. If data of any block is changed, the hash value of that block would not match with the previous block's hash and hence that would indicate that the data is tampered[5].

1.3 Problem definition

To design and develop student marksheet validation system using smart contracts using Etherium framework and to analyze the performance of this system considering throughput.

1.4 Application

- Student can download the respective marksheets.
- Companies to verify the document.
- Documents directly used for higher education.

1.5 Objectives and Scope of the project

1.5.1 Objectives

- To setup Blockchain network.
- To create smart contract for storing marksheets.
- To integrate IPFS with Blockchain Technology.
- To perform analysis with the exisiting and proposed method considering throughput.

1.5.2 Scope of the project

- Option to upload documents in blockchain for each student.
- To increase the security of the system, decentralized application can be developed using ethereum.
- University able to add the student marksheet.

2 Requirement Analysis

The following contains the system requirements of our proposed model which include the functional and non-functional requirements.

2.1 Functional Requirements

1)University shall be able to store the students details.
2)Verifier shall be able to register for University page.
3)Student shall be able to view the documents.
4)Student shall be able to download the marksheets.
5)Verifier shall be able to upload the document to verify them.

2.2 Non-Functional Requirements

1)The transaction recorded should be 100 percent consistent
2)Registration of verifier should be less than 0.2sec
3)Login must be successfull of university, student and verifier should take less than 0.2sec

2.3 Software Requirement

1)Ethereum Framework
2)Geth version 1.10.16 stable
3)Visual Studio Code-Code Editor

2.4 Hardware Requirement

1)System with 4GB RAM
2)1TB Hardrive

3 System Design

3.1 Architecture Design of the system

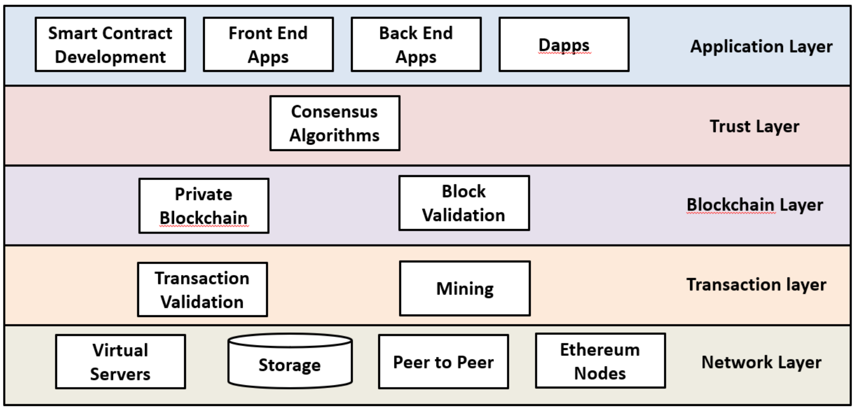


Figure 1: Layered Architecture of the Blockchain

As shown in Fig. 1, the Blockchain's technological components include transactions, blocks, consensus, applications, and smart contracts. All of these components are organised into layers. P2P network containing Ethereum nodes is referred to as the network layer. Transactions triggered by users or smart contracts are referred to as the transaction layer. The trust layer refers to the consensus procedure for block and transaction validation, whereas the blockchain layer refers to connected blocks containing all transaction information. Applications and smart contracts are referred to as the application layer.

3.2 Usecase Diagram

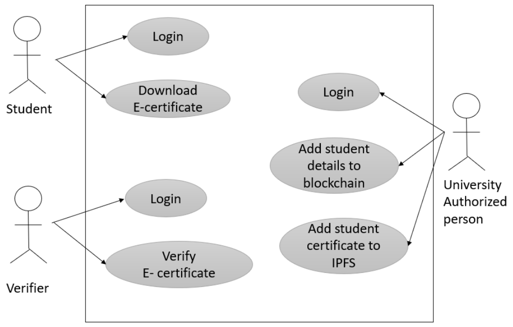

Student
Login
Download E-certificate
Login
Add student details to blockchain
University Authorized person
Verifier
Login
Verify E- certificate
Add student certificate to IPFS

Figure 2: Usecase Diagram

The key components of our research, as well as its operating mechanism, are depicted in Figure 2. The following are the most important architectural elements: The ledger is maintained by a node, which can be a university or a user. Transaction: In the blockchain architecture, a transaction is a small unit of building block. A block is a data structure that is used to keep track of transactions that are broadcast to all network nodes. A chain is a grouping of blocks.Use case diagram The fig 2 shows the core components of our project and its working mechanism. The main core components of architecture are: Node: university or user is considered as node maintains the ledger. Transaction: A small unit of building block which used in the blockchain architecture. Block: A block is a data structure used for maintaining the transactions which are broadcasted to all the nodes of network. Chain: A sequence of blocks. Here student can login and download E-Certificate and University authorized person can login ,add student details and add student certificate where as verifier can login and also download ,view the E-Certificate for validation purpose

3.3 Proposed Solution

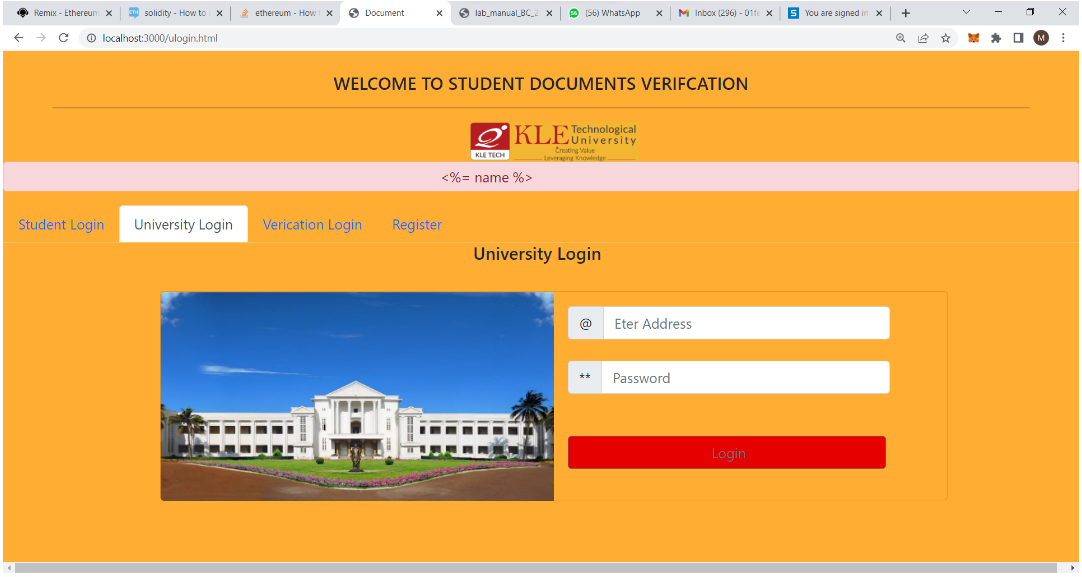
In the proposed solution, we have used blockchain technology to store the mark sheets and certificates instead of manually storing them on the centralised storage. In its most basic form, blockchain is a type of shared database that varies from traditional databases in the way it keeps data. Data is stored in blocks on the blockchain, which are connected together using the cryptographic principle. The blockchain's purpose is to enable for the recording and distribution of digital data without the ability to modify it. In this approach, a blockchain serves as the foundation for immutable ledgers, or transaction records that cannot be changed, deleted, or destroyed.

The benefits of blockchain are as follows:
1) Decentralized structure: The blockchain provides a distributed ledger, which means that every node in the network will have a copy of the ledger; however, the ledger will not be controlled by a central authority.
2) Increased security and privacy: Blockchain uses cryptographic concepts to guarantee complete security and privacy.
3)Visibility and traceability: The blockchain provides a system for data visibility and traceability.
4) Immutability: Blockchain is immutable in nature, which means that once data is written into the blockchain, no one can erase or change it.

We apply the major characteristics of the blockchain in this marks sheet validation system, which include immutable ledger, tracability, distributed ledger, privacy, and security. We are proposing marks cards in the form of pdf based on student information. The difficulty with storing pdf on the blockchain is a phenomena known as latency. The most cost-effective method is to keep the complete pdf int ipfs and store the document hash in the blockchain ledger.

As for the implementation, we used go-ethereum(geth) to build up the blockchain network, then integrated ipfs with geth to calculate the hash of the pdf using the sha 256 crptographic hash function. In the blockchai, saving the pdf hash value computed by sha 256 and the ipfs address.

3.4 User Interface Design

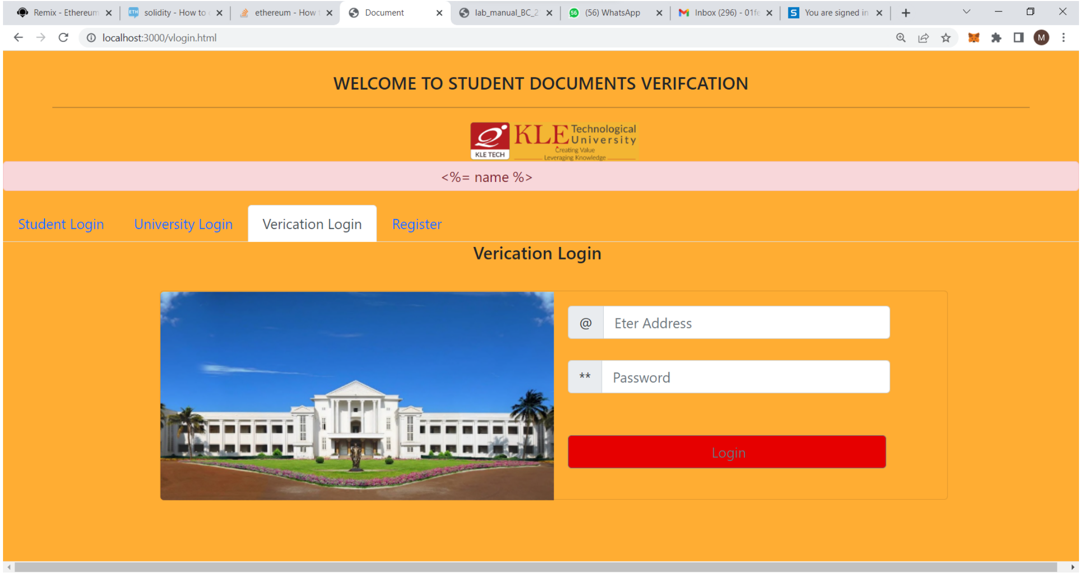


localhost:3000/ulogin.html

WELCOME TO STUDENT DOCUMENTS VERIFCATION

<%= name %>

Figure 3: University Login

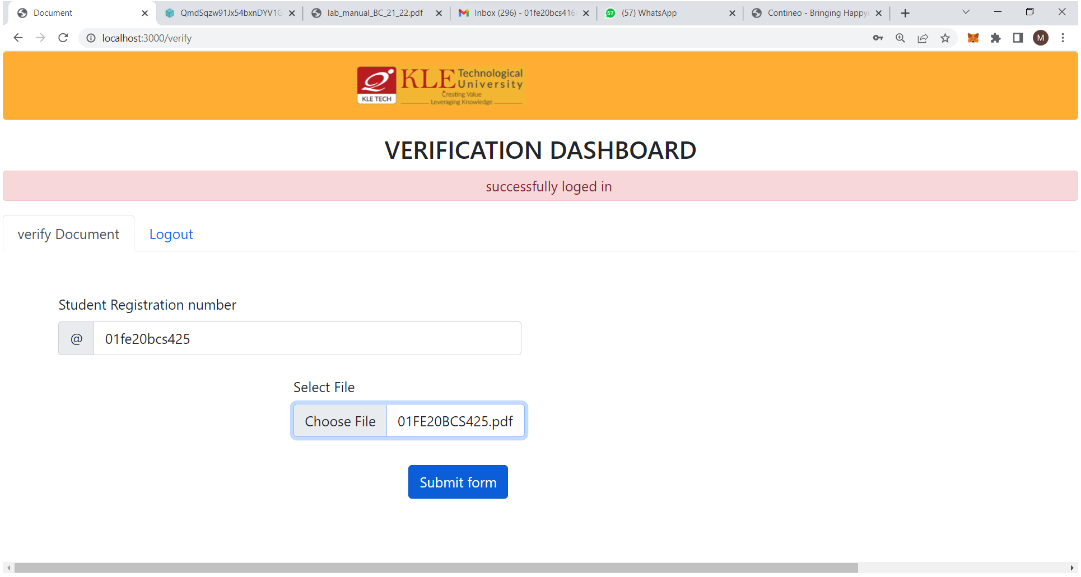


localhost:3000/vlogin.html

WELCOME TO STUDENT DOCUMENTS VERIFCATION

<%= name %>

Figure 4: Verification Login



VERIFICATION DASHBOARD

successfully loged in

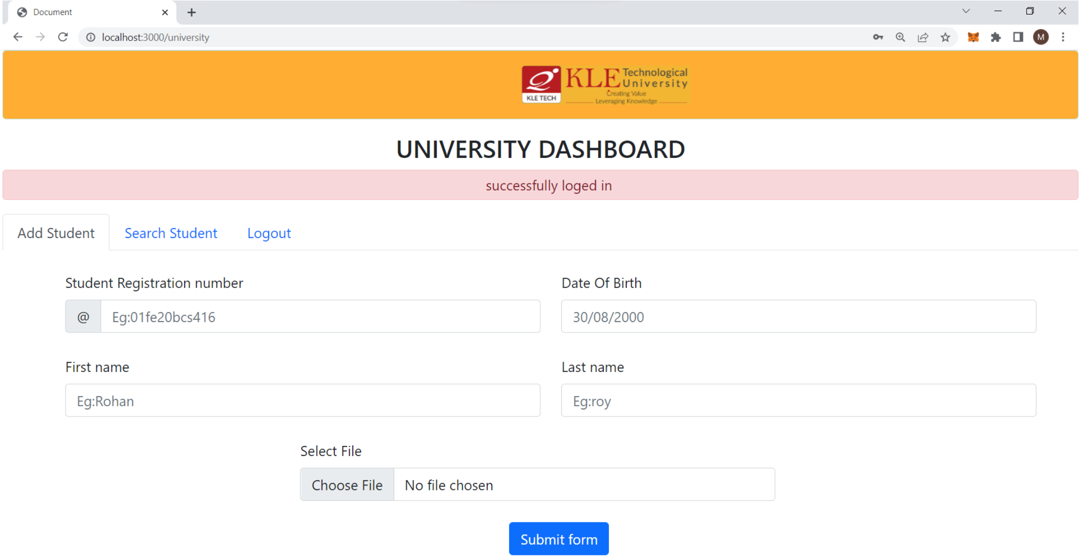
Student Registration number

@ 01fe20bcs425

Select File

01FE20BCS425.pdf

Figure 5: Verification Dashboard



UNIVERSITY DASHBOARD

successfully loged in

Student Registration number

@ Eg:01fe20bcs416

Date Of Birth

30/08/2000

First name

Eg:Rohan

Last name

Eg:roy

Select File

No file chosen

Figure 6: University Dashboard

4 Implementation

4.1 Proposed Methodology

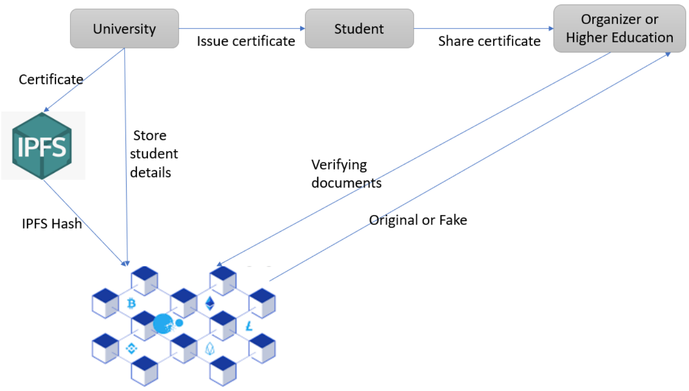


Figure 7: Proposed System

The hash will be created depending on the certificate, as shown in the diagram. When the organisation hash is checked against the server database, if the hash matches the server database entry, the certificate is genuine; otherwise, it is a forgery.

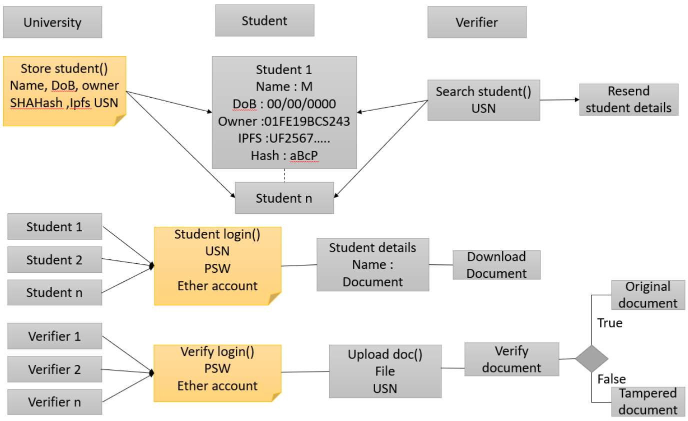


Figure 8: Process Diagram

The university, student, and verifier variables make up the digital marksheet validation system. Before storing the document, the variables' values are set. The storestudent function is used to store student information for the purpose of authenticating marksheets. Students can log in and view or download their grades. Verifiers must log in with their password and ether account and be able to upload the student's certificate for verification. If the hash of the uploaded marksheets matches, it is genuine; otherwise, it is a forgery.

4.2 Modules

4.2.1 Blockchain

The blockchain can be thought of as an immutable database that underpins the entire project. It establishes a secure environment in which actions are visible and unaffected.

4.2.2 Ethereum

Ethereum is a decentralised open-source Blockchain with smart contract capabilities. Ethereum is the best example of Blockchain because it is the most widely used and second most expensive cryptocurrency system after bitcoin.

4.2.3 SmartContract

A smart contract is a piece of code that executes on a Blockchain when a user takes an action. A smart contract can be written in a variety of languages, from low-level languages like C++ and Java to high-level languages like Solidity, which is very similar to Typescript.

4.2.4 Solidity

Solidity is a smart contract programming language based on object-oriented programming. It's used to build smart contracts on a variety of Blockchain systems, the most popular of which is Ethereum. It's very similar to Typescript, however it has a lot more data types.

4.2.5 Ethash

The proof-of-work function of Ethereum-based Blockchain currency is called Ethash. It is a hash function that belongs to the Keccak family, which also includes the SHA-3 hash functions. Ethash, on the other hand, is not a SHA-3 function and should not be mistaken with it.

4.2.6 IPFS

The issue with storing all of the data on the blockchain is a phenomenon known as latancey. This simply refers to the time it takes network users to upload or download files, such as advertisements. The most economical way is to keep the entire page while storing the hash of the documents in the blocks. We're talking about storage systems like the Interplanetary File System, for example (IPFS). This entails dividing files into many parts and storing them in multiple locations on system members' machines.

This method provides a number of advantages.
1) The user will only download the file if they are interested.
2) This is a peer-to-peer system.
3)Data loads faster due to the increased bandwidth.

We are integrating an ipfs client for storing student marks cards in this marks sheet validation system, and the address of the marks sheet will be maintained on the blockchain provided by ipfs.

4.2.7 SHA256

The SHA256 algorithm is a cryptographic concept that is used to encrypt sensitive readable material into an unreadable format. This SHA256 hash function takes the string's arbitary length and converts it to a fixed length hash value. A mathematical procedure generates a string of numbers and letters, which is the hash value. The function will run for 80 cycles before returning the final 256-bit hash value.

Following are some of the benefits of SHA256:
1) It is deterministic in the sense that it always returns the same hash value for a given input (or file).
2) Collusion is unlikely since various files are unlikely to have the same hash value.

3) A hash can be computed in a short amount of time

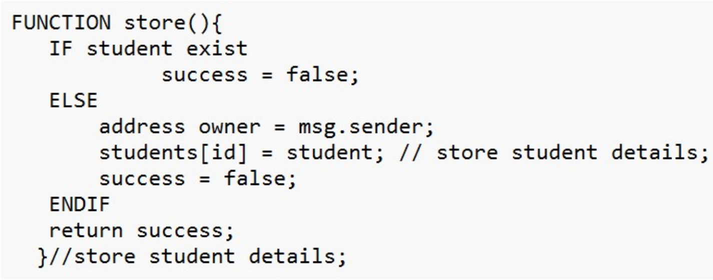
We employed the SHA256 ciptographic hash function to generate the hash value for validation in this marks card validation system, and that hash value will be kept on the blockchain ledger.

4.3 Pesudo code

Smart contract's goal is to save and retrieve student information and verification information based on their ether address and student registration. Once the student information is stored, no one can change it on the blockchain, which is one of the smart contract's simplest procedures.

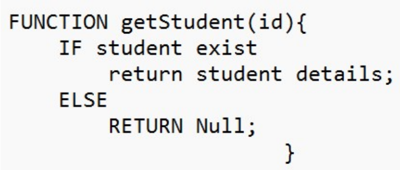
The pseudo code below is for the store function, which stores student records on the blockchain using the Solidity programming language. The parameters for the store function are the following: student register number, first name, last name, date of birth, hash value of the marks card, and ipfs unique path.

The function first determines whether the student is present or not; if the student record is present on the blockchain, the function will not store the student record and will return a false message; if the student record is not present on the blockchain, the function will successfully store the student record on the blockchain system and will return a true message.



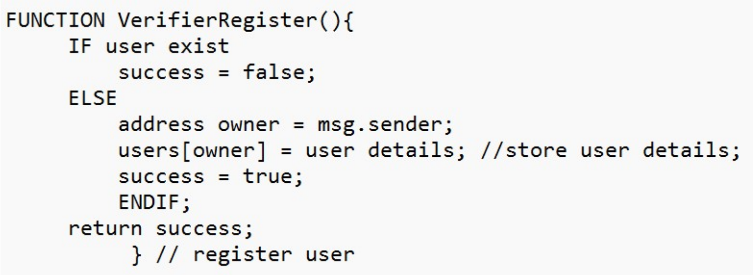
```
FUNCTION store(){
   IF student exist
            success = false;
   ELSE
       address owner = msg.sender;
       students[id] = student; // store student details;
       success = false;
   ENDIF
   return success;
  }//store student details;
```

The getStudent function, which retrieves student records from the blockchain ledger, is demonstrated in the pseudo code below. The student register number is passed to the getFunction function. If the student records exist on the blockchain, the function will search the student details and return all student information associated with the student register number; if the student records do not exist on the blockchain, the function will return NULL from the blockchain system.



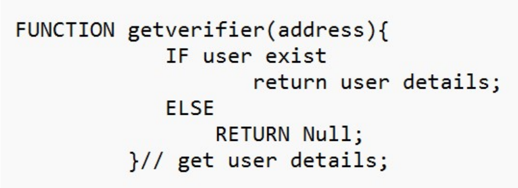
```
FUNCTION getStudent(id){
    IF student exist
        return student details;
    ELSE
        RETURN Null;
                     }
```

The verifierRegister function, which keeps verifier details on the blockchain, has the pseudo code below. The following are the parameters for the verifier-Register function: ether account, first name, last name, and account password The function initially determines whether or not the verfeir accounts are available on the blockchain; if they are, the method will not create new accounts. If the verifier account is not existent on the blockchain, the function will successfully register the verifier account on the blockchain system and provide a true message.



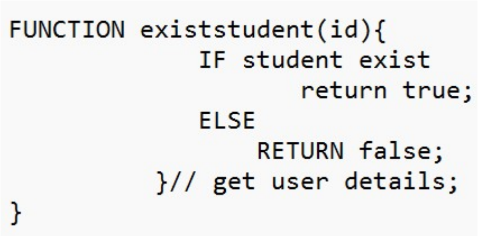
```
FUNCTION VerifierRegister(){
     IF user exist
         success = false;
     ELSE
         address owner = msg.sender;
         users[owner] = user details; //store user details;
         success = true;
         ENDIF;
     return success;
          } // register user
```

The getverifier function, as seen in the pseudo code below, pulls verifier account details from the blockchain ledger for account validation. The ether address is passed as a parameter to the getVerifier function. If the account details exist on the blockchain, the function will provide all of the verifier's information; if the account details do not exist on the blockchain, the function will return NULL from the blockchain system.

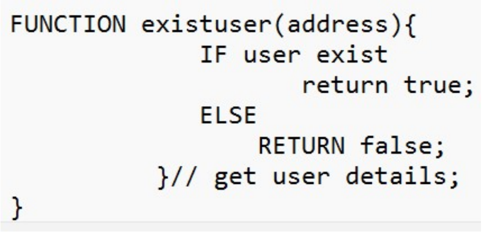


```
FUNCTION getverifier(address){
            IF user exist
                   return user details;
            ELSE
                RETURN Null;
         }// get user details;
```

The existuser and existStudent functions in the pseudo code below verify the state of student and verifier on the blockchain ledger. If both details are found on the blockchain, the relevant function will return true massage; if neither detail is found on the blockchain, the functions will return false.



```
FUNCTION existstudent(id){
             IF student exist
                    return true;
             ELSE
                 RETURN false;
          }// get user details;
}
```



```
FUNCTION existuser(address){
             IF user exist
                    return true;
             ELSE
                 RETURN false;
          }// get user details;
}
```

5 Results and Discussions

5.1 Test cases

5.1.1 Compatibility Testing

Test Case Id	Input Description	Expected Output	Actual Output
1	Geth Platform	Compatible	Compatible
2	Remix IDE	Compatible	Compatible

5.1.2 Component Testing

Test Case Id	Input Description	Expected Output	Actual Output
1	StudentInfo Contract	StudentInfo Contract deployed	StudentInfo contract deployed.

5.2 Test plan

5.2.1 Test cases

Test Case Id	Input Description	Expected Output	Actual Output
1	University Login	Accept university a/c to login	Accept university a/c to login
2	University Login	Invalid university a/c to login	Invalid login
3	Uploading document by university a/c	Successfully uploaded	Successfully uploaded
4	Student Login	Accept Student a/c to login	Accept Student a/c to login
5	Student Login	Invalid Student a/c to login	Invalid Student a/c to login

5.3 Performance analysis

Number of transactions	Processing time with IPFS(seconds)
10	0.304
25	0.650
50	1.053
75	1.288
100	1.401
125	1.578
150	1.661

Table No :01

The table interprets the performance analysis of Geth considering the processing time with number of transactions executed by the system.

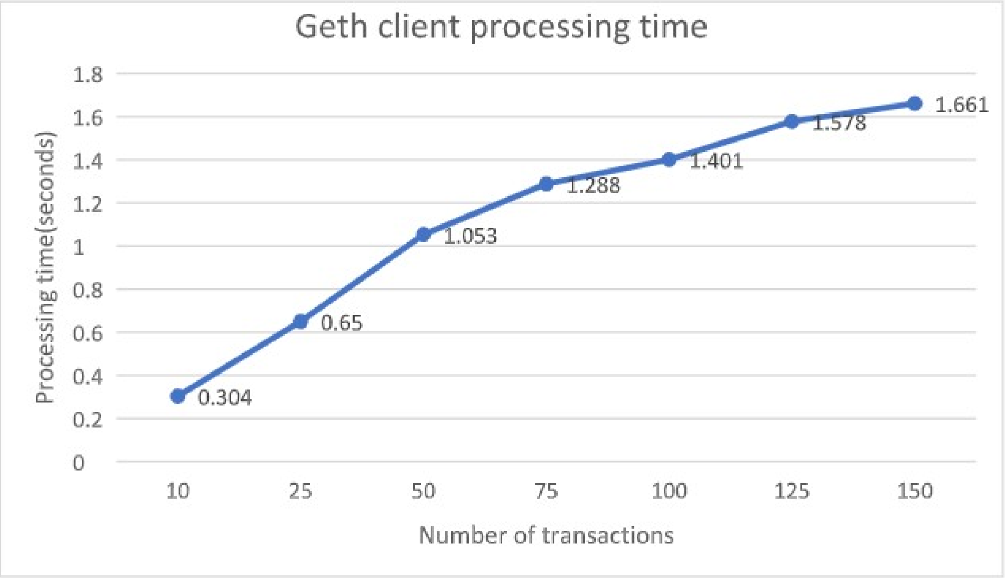
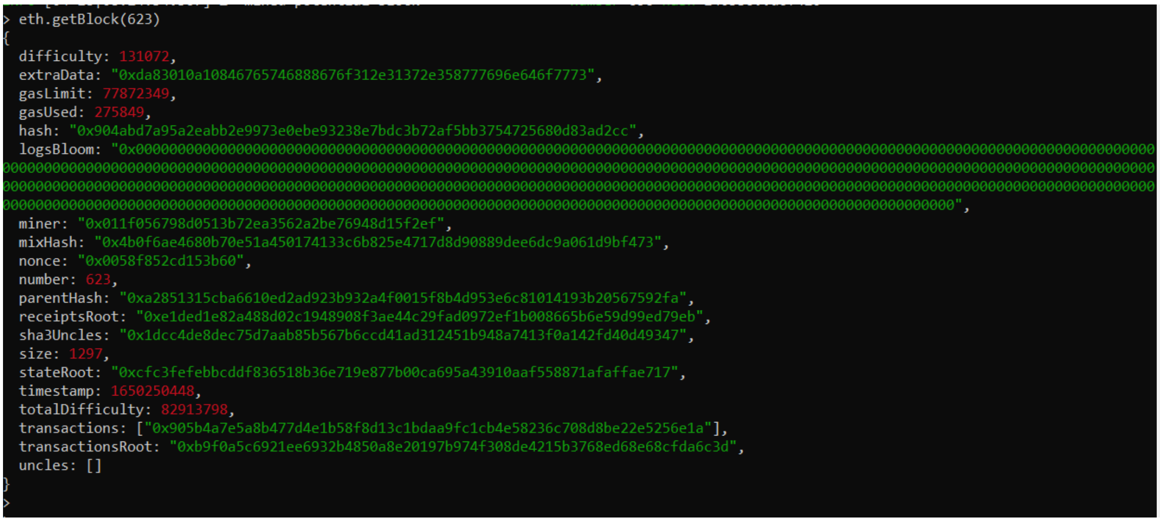


Figure 9: The processing time considered by Geth client

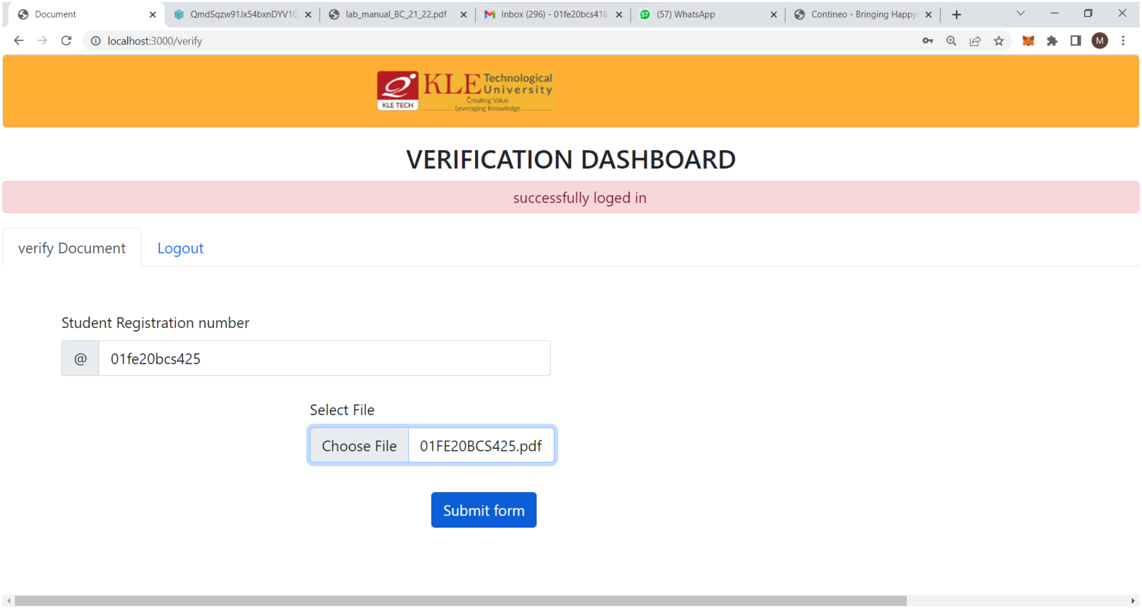
According to given figure 9: it shows number of transactions against processing time(sec) where as number of transaction are increases the processing time corresponding also increases and number of transactiosns more than 75 or 100 the processing time will be nearly same.

5.4 Snapshots



```
> eth.getBlock(623)
{
  difficulty: 131072,
  extraData: "0xda83010a10846765746888676f312e31372e358777696e646f7773",
  gasLimit: 77872349,
  gasUsed: 275849,
  hash: "0x904abd7a95a2eabb2e9973e0ebe93238e7bdc3b72af5bb3754725680d83ad2cc",
  logsBloom: "0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
  miner: "0x011f056798d0513b72ea3562a2be76948d15f2ef",
  mixHash: "0x4b0f6ae4680b70e51a450174133c6b825e4717d8d90889dee6dc9a061d9bf473",
  nonce: "0x0058f852cd153b60",
  number: 623,
  parentHash: "0xa2851315cba6610ed2ad923b932a4f0015f8b4d953e6c81014193b20567592fa",
  receiptsRoot: "0xe1ded1e82a488d02c1948908f3ae44c29fad0972ef1b008665b6e59d99ed79eb",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 1297,
  stateRoot: "0xcfc3fefebbcddf836518b36e719e877b00ca695a43910aaf558871afaffae717",
  timestamp: 1650250448,
  totalDifficulty: 82913798,
  transactions: ["0x905b4a7e5a8b477d4e1b58f8d13c1bdaa9fc1cb4e58236c708d8be22e5256e1a"],
  transactionsRoot: "0xb9f0a5c6921ee6932b4850a8e20197b974f308de4215b3768ed68e68cfda6c3d",
  uncles: []
}
>
```

Figure 10: Details stored in block


Document
QmdSqzw91Jx54bxnDYV1C
lab_manual_BC_21_22.pdf
Inbox (296) - 01fe20bcs41
(57) WhatsApp
Contineo - Bringing Happy
localhost:3000/verify
KLE Technological University
Creating Value Leveraging Knowledge
KLE TECH

VERIFICATION DASHBOARD

successfully loged in

verify Document Logout

Student Registration number

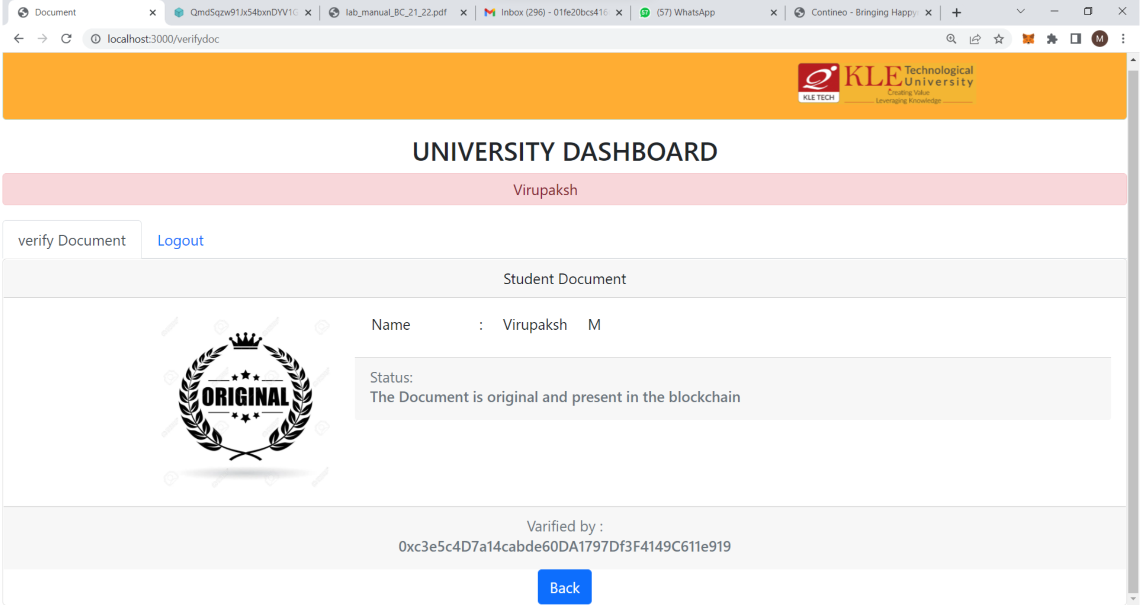
@ 01fe20bcs425

Select File

Choose File 01FE20BCS425.pdf

Submit form

Figure 11: Verification Dashboard



UNIVERSITY DASHBOARD

Virupaksh

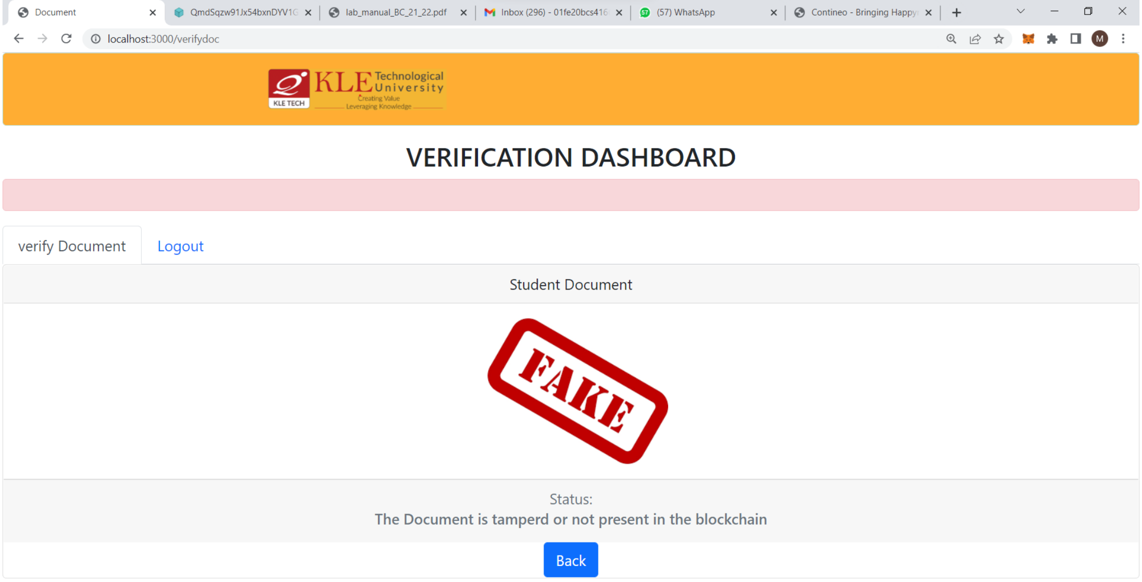
Student Document

Name : Virupaksh M

Status:
The Document is original and present in the blockchain

Varified by :
0xc3e5c4D7a14cabde60DA1797Df3F4149C611e919

Figure 12: Document is Original



VERIFICATION DASHBOARD

Student Document

Status:
The Document is tamperd or not present in the blockchain

Figure 13: Document is Fake

6 Conclusion and Future Scope

6.1 Conclusion

One of the key values of Blockchain is the creation of immutable ledgers. This characteristic aids us in creating a system in which all processes are transparent and immutable. Our system automates the process of verifying certificates and decreases the amount of manual work required to do so. Students also have a low chance of losing their certificate. The certificate's hash is kept in the blockchain, while the original document is kept in the Inter Planetary File System (IPFS). According to the performance analysis, any amount of records can be kept in blockchain, and the system will be able to access them in constant time. and for a specific set of records, uploading will take around the same amount of time.

6.2 Future Scope

The propsed system is able to store the hash value in blockchain of only single document but in future system method can enhance the project to upload and store the multiple document of single student.To increase the performance of the system considering throughput and scalability more than 5 percent by using parity network.In proprosed solution document can be stored to a student this can be enhanced to Multiple documents of single student can be stored. studying in-detail the performance of transaction can lead to generation to new ethereum client which can further reduce the limitations on faster blockchain.

References

[1] Mustafa A Ali and Wesam S Bhaya. Higher education's certificates model based on blockchain technology. In *Journal of Physics: Conference Series*, volume 1879, page 022091. IOP Publishing, 2021.

[2] Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen. Blockchain and smart contract for digital certificate. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 1046–1051, 2018.

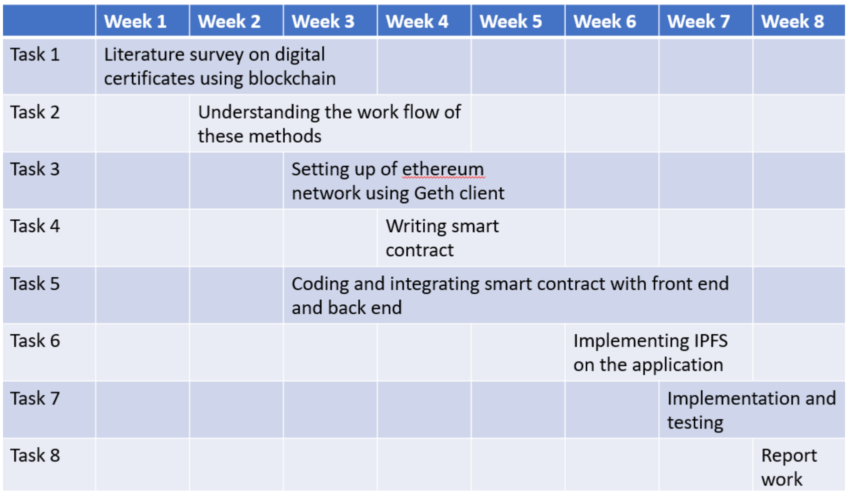
[3] Ravi Singh Lamkoti, Devdoot Maji, A Bharati Gondhalekar, and Hitesh Shetty. Certificate verification using blockchain and generation of transcript. *Int. J. Eng. Res. Technol*, 10(3), 2021.

[4] Qiuhong Zheng, Yi Li, Ping Chen, and Xinghua Dong. An innovative ipfs-based storage model for blockchain. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 704–708, 2018.

[5] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564, 2017.

7 Appendix

7.1 Gantt Chart



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Task 1	Literature survey on digital certificates using blockchain							
Task 2		Understanding the work flow of these methods						
Task 3			Setting up of ethereum network using Geth client					
Task 4				Writing smart contract				
Task 5			Coding and integrating smart contract with front end and back end					
Task 6						Implementing IPFS on the application		
Task 7							Implementation and testing	
Task 8								Report work

Figure 14: Gantt Chart