International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 03, March-2021

Certificate Verification using Blockchain and Generation of Transcript

Ravi Singh Lamkoti
Dept. of Information Technology
Vidyavardhini's College of Engineering and Technology
Vasai, Palghar

Devdoot Maji
Dept. of Information Technology
Vidyavardhini's College of Engineering and Technology
Vasai, Palghar

Hitesh Shetty
Dept. of Information Technology
Vidyavardhini's College of Engineering and Technology
Vasai, Palghar

Prof. Bharati Gondhalekar
Asst. Prof. Dept. of Information Technology
Vidyavardhini's College of Engineering and Technology
Vasai, Palghar

Abstract— **An estimated 26.3 million students enrolled in Indian higher education in 2018-19 and nearly 9 million graduates annually. A student, whether he is a high school student or an undergraduate, or maybe a postgraduate, generates a lot of certificates that may include results, diplomas, or transcripts during this entire duration of studies. For admission, students need to produce these certificates in institutions or companies. Tracking these certificates and validating their authenticity manually becomes a tedious job. The absence of an appropriate anti-forge system leads to a scenario where it is found that the graduation certificate is forged. To make the data more secure and safe, everything needs to be digitalized with the principle of Confidentiality, Reliability, and Availability. All of these can be achieved with a technology named Blockchain. Briefly, the flow of our system will consist of a Certificate issuer who will generate certificates and those certificates will be validated by a panel within that organization before being sent to a student. Each certificate will have a unique hash key which can be used to validate the authenticity of the certificate by any organization through the portal. The benefit of such a system is that the student also faces less risk of losing or damaging a certificate and the validation of the certificate can also be done quite easily.**

Keywords— Digital Certificates, Blockchain, IPFS, Certificate Generation and Verification.

I. INTRODUCTION

In India, the basic structure of a student's studies goes like taking admission in kindergarten, after that changing of school for primary, secondary, and high school studies. Now, after completing high school students, need to get admission into junior college. For graduation, there's also once again changing of college. This is the basic cycle for student's study years. After this, some students continue to pursue higher studies.

So the problem with this cycle is that a student needs to produce all his certificates in each stage for validation. This poses a risk of losing and damaging the certificate. And it is tedious for the validator to authenticate each certificate.

With such a huge population in our country, almost every year 26.3 million students graduate. It is very hard to keep track and validate such a huge amount of records. Due to this, an unwanted scenario rises i.e. tampering and production of fake or duplicate certificates. There are a lot of hidden agencies in our country who are running this scam behind everyone's back. Technology has moved quite forward until now. Distinguishing between a fake and an original certificate will require a lot of concentration and result in wastage of precious time.

For removing this disadvantage, a technology named Blockchain comes into our life as a savior. So why to use Blockchain? Because the data in a Blockchain cannot be changed under realistic conditions. Even if data is changed, it just takes a second to let us know about the tampering. In Blockchain a data or a node is validated only when multiple parties approve it. So, the system would be Reliable and Authenticated at any instance of time.

Now, the issue of tampering is solved. The next issue that comes into the picture is time consumption for validation. The system that we will be building will not only validate the certificates but also generate certificates. So it is like killing two birds with one stone. As everything is automated, it takes mere seconds to validate the document.

Since everything will be stored digitally, a student doesn't need to worry about losing or damaging the certificate in the process of validation. This proposed system not only removes the loopholes in our current system but also gives us an effective and concrete solution.

II. LITERATURE SURVEY

The project focuses on building an immutable certificate generation as well as a validation system. For this, we have referred few previously published papers and works of the various individual in this field. Our Literature Survey mainly focused on Blockchain Technology, an advanced Storage System, and Digital Certificate Validations.

Our first paper was titled, An Overview of Blockchain Technology [1] which provided in-depth knowledge regarding Blockchain. It introduced various terms regarding this technology and the most important concept called a smart contract. In the Blockchain, the hash of the data is stored in its preceding block, and it forms a long chain of nodes. If data is changed, its hash will change, and it won't match with the hash value stored in the previous block and hence letting us know about the tampering of data.

The second paper was titled, Blockchain and Smart Contract for Digital Certificate [2]. Their design consisted of 3 actors. First, there were institutions, second being the students, and last the service provider. The drawback of their method was that they were using 'one hash as a key', which makes it publicly accessible once they have the hash.

Moving onto our next paper, Tamper Proof Birth Certificate [3]. Similar to the second paper their design was almost the same except they were running the AES algorithm and storing the data in the IPFS. Their system was explicitly for Birth Certificates. The drawback that was seen was that the original document was never stored anywhere nor it had the functionality to generate the certificates online.

For solving the issue of document storage, we researched a separate paper which was titled, BlockIPFS (Blockchain-enabled Interplanetary File System for Forensic and Trusted Data Traceability) [4]. This paper introduced us to the knowledge of IPFS and how to integrate it with Blockchain. They compared traditional IPFS with Blockchain with IPFS, and the results displayed that BlockIPFS won in most categories, such as upload transaction, read transaction, and Download transactions.

The Final Paper was a Blockchain-Based Identity Verification Model [5]. Similar to the second and third paper their system consisted of an Issuing Authority who will generate the document, a hashing algorithm works over it, and its value is stored. Since other systems had public hash keys, they increased security by using asymmetric encryption.

III. PROPOSED METHODOLOGY

A. Modules

1) Blockchain: Blockchain can better be understood as an immutable database and laid the foundation of the whole project. It provides a trusted environment where actions have done are visible and can't be tampered with.

2) Ethereum: Ethereum is a decentralized open-source Blockchain featuring smart contract functionality. Ethereum is itself the best example of Blockchain and its a cryptocurrency system which is the most widely used and is the next expensive cryptocurrency after bitcoin.

3) SmartContract: Smart contracts piece of code that runs on a Blockchain when a user performs some action. A Smart contract is written in many different languages including low-level languages like C++, Java, and high-level languages like Solidity which is closely similar to Typescript.

4) Solidity: Solidity is an object-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various Blockchain platforms, most notably, Ethereum. It is closely similar to Typescript but with more specific data types.

5) Ethash: Ethash is the proof-of-work function in Ethereum-based Blockchain currencies. It is a hash function belonging to the Keccak family, the same family to which the SHA-3 hash functions belong to. However, Ethash is not an SHA- 3 function, and should not be confused with them.

6) IPFS: The InterPlanetary File System is a peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices.

7) Metamask: MetaMask is an extension for accessing Ethereum enabled distributed applications or "Dapps" in your browser. The extension injects the Ethereum web3 API into every website's javascript context so that Dapps can read from the blockchain.

8) Ganache: Ganache is used for testing Solidity contracts on a personal Ethereum Blockchain. It by default provides an easy setup for spinning up a network with around ten users with each having 100 eths on their account. These accounts can be used to mimic the transactions between the users.

9) Truffle: Truffle provides easy compilation, linking, deployment, and binary management of smart contracts written in solidity language.

10) Rinkeby: Rinkeby provides an Ethereum test network which is used by developers to test their code and perform some actions. It just provides an environment that works on proof of authority, unlike ethereum which works on proof of work.

11) Node JS: Node JS is used to write backends and is responsible for serving frontend pages, assets and managing user authentication using JWT(Json Web Token). It also has web3 as a dependency which allows us to run solidity code on frontend.

12) React: React is used to write our frontend and serves a purpose of providing a better user experience in the frontend for the end user as it provides functionality like no page reload on page switch and fast loading of sites. Keeping security in mind we have added Next.js which compiles and saves html pages in the backend and provides fast user experience along with better SEO for react pages and security as it doesn't reveal any backend details on the user end.

13) MongoDb: MongoDB is the type of NoSql database and is used to store details about Users and their authentication saved in an encrypted format and about their session details.

B. Project Description

For the loopholes in our current methodologies, we have proposed our system which will automatically generate certificates as well as validates them. The data will be authenticated, reliable and unchangeable. The entire process is thoroughly described in the upcoming sections.

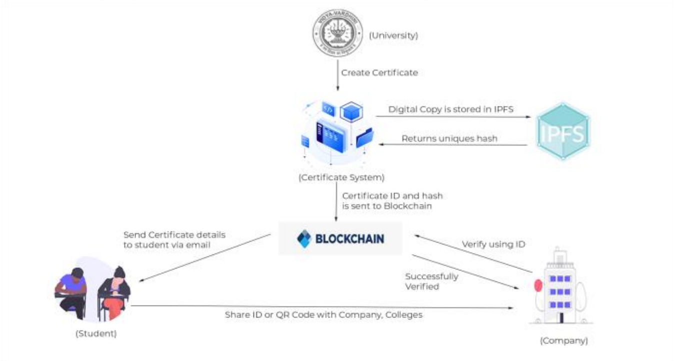

(University)
Create Certificate
Digital Copy is stored in IPFS
Returns uniques hash
IPFS
(Certificate System)
Certificate ID and hash is sent to Blockchain
BLOCKCHAIN
Send Certificate details to student via email
Verify using ID
Successfully Verified
Share ID or QR Code with Company, Colleges
(Student)
(Company)

Fig 1. Workflow of the System

1) *System Design and Working*

Our proposed solution offers ease of determining whether a document is authentic or not and also checking the integrity and originality with the help of distributed technologies like IPFS and Ethereum smart contracts. The above figure shows how users interact with smart contracts. The participants which interacts with our smart contract are as follows:

a) College: College acts as a Certificate issuing authority. This entity can be any organization that wants to issue a certificate. College or Certificate issuing authority only has the right to issue one or more certificates in the system. It will also include a phase where the College will generate certificates after validating and authorizing the details.

b) Student: Students will be able to download and view digital documents. Students also receive the hash regarding the document issued for him, which in the future can be used to access the document and verify it.

c) Company: Company will be the user who will have access to regarding originality, authenticity, and integrity of the documents with the help of the digital signature of the document.

Initially, the college i.e. the Certificate issuing authority will feed in the Certificate Template. After that, the issuer will fill in the details required to generate a document. Grabbing those details, the data will be appended to a predefined certificate template.

After that, a preview of the document will be generated. If the Issuer feels the document is perfect, he will click on the 'Approve' button.

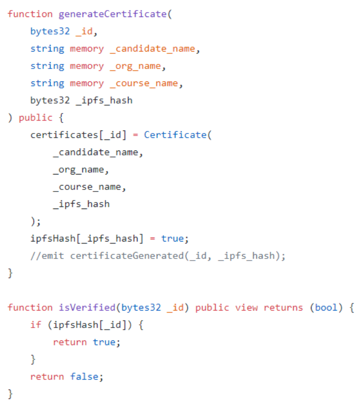
After clicking on the Approve Button and storing the data, there are a lot of things that happen. Let us see what actually happens. First, the document data is gathered, and it is appended in a bit array. Now IPFS is given this data, and it applies its hashing algorithm. This hash generated is stored in the IPFS along with the original Document.

IPFS now passes this data to the Blockchain. Now the issuer needs to approve the generation charges in Metamask. After that, this hash is stored in the Blockchain and cannot be changed under normal circumstances. Even if by any chance the data is changed, the other nodes in the Blockchain will notify.

Now the Student can send this hash or his digital certificate on his certificate to various organizations. The issuer can either upload this document or type in the hash key. The System will respond whether the document was legit or not.

IV. IMPLEMENTATION DETAILS

The backbone of any blockchain project is its contract, contract is a code that runs on an ethereum node. This code is written in Solidity Language which is a high-level language which is derived from Javascript and strongly typed languages and is mainly used to write contracts. The code mostly contains two contracts file Migration.sol and *'Certification.sol'*. The migration.sol is the default solidity file that comes along with ethereum setup and contains code about how transactions should happen and keeping track of transactions. The Certification.sol file contains the main code of the system and consists of '*generateCertificate()*' and '*isVerified()*'. generateCertificate takes information and details which are to be included in the certificate. The isVerified() function takes a hash as an input which it uses to check if the hash is associated with any document or not. If the function will be original the function will return boolean value True denoting the hash is associated with a verified block else will return false which lets the user know that the document has been tampered.



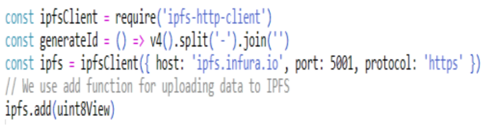
```
function generateCertificate(
    bytes32 _id,
    string memory _candidate_name,
    string memory _org_name,
    string memory _course_name,
    bytes32 _ipfs_hash
) public {
    certificates[_id] = Certificate(
        _candidate_name,
        _org_name,
        _course_name,
        _ipfs_hash
    );
    ipfsHash[_ipfs_hash] = true;
    //emit certificateGenerated(_id, _ipfs_hash);
}

function isVerified(bytes32 _id) public view returns (bool) {
    if (ipfsHash[_id]) {
        return true;
    }
    return false;
}
```

Fig 2. Storing IPFS Hash on Blockchain

The Certification.sol file contains a struct Certificate which will be the structure of our data block. This provides a blueprint for the data to be stored in the ethereum block. This contains a byte32 ipfs_hash which is associated with the generated document and will be used further for verifying the originality of the document. Others are the details needed to be included in the Certificate.

These solidity files are once written can be compiled on platforms like Remix IDE online which understands how to compile Solidity. Here Truffle compiles the solidity and converts the given solidity files to something called abis. This contains the machine code of the solidity functions and based on the computation and space consumption, the cost of the transaction is determined. These files will generally after compilation and execution have the same name as the solidity file name but with JSON extension. This JSON file is then used to create a web3 instance of executing solidity code on ethereum blockchain.

The next most important part of the system is IPFS(Inter Planetary File System) which is used to upload documents. Since it's distributed in nature and no single user stores the document the uploaded documents are secured and can be accessed with the help of hash. Each node in the IPFS system will have a part of the file hence no one can have access to a document at a time.



```
const ipfsClient = require('ipfs-http-client')
const generateId = () => v4().split('-').join('')
const ipfs = ipfsClient({ host: 'ipfs.infura.io', port: 5001, protocol: 'https' })
// We use add function for uploading data to IPFS
ipfs.add(uint8View)
```

Fig 3. Adding generated Document to IPFS

V. RESULTS AND DISCUSSIONS

Following are some of our implementation results:

A. *Registration-*

A user, whether he is a student or teacher, or validator needs to register themselves first on the portal. The teacher

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 03, March-2021

i.e. the Certificate issuer will only be able to generate certificates. The Student can download his/her Document, and the Validator will be able to validate and view the document. So each actor has their own features and doesn't have access to the features they do not need.

Once a User is registered he/she will now be able to access the system from the login window. If the credentials entered match the registration credentials which was stored in MongoDB, the user will be logged in, or else an error will be thrown.

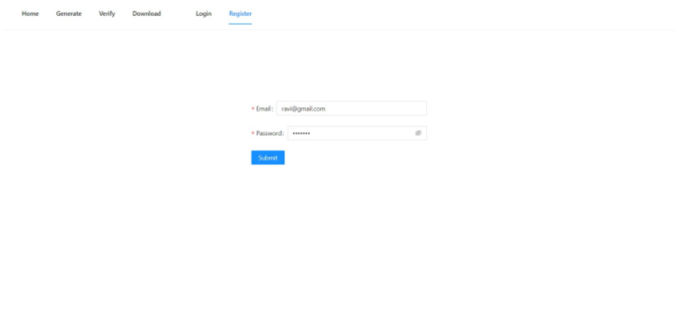


Fig. 4. Registration Page

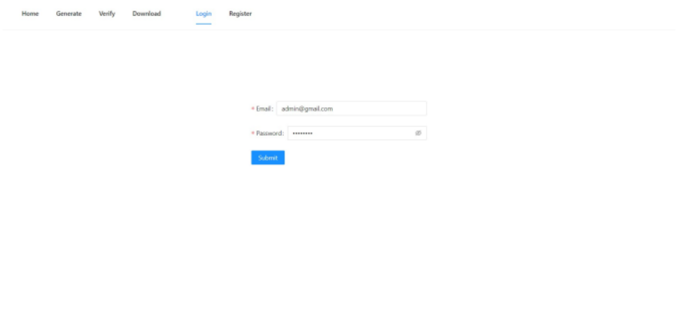
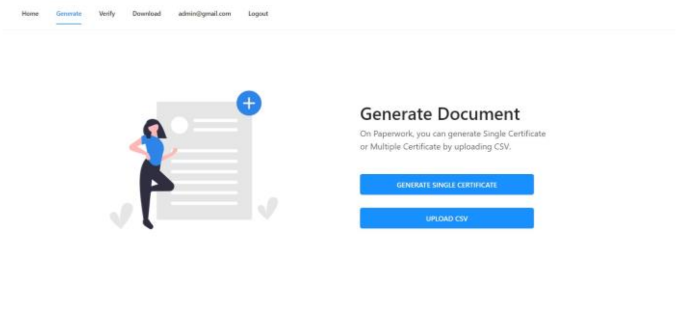


Fig. 5. Login Page

B. *Certificate Issuer-*

The Certificate issuer has two options to generate certificate. Either to fill in details in a form to generate a single certificate or to upload a CSV to make certificates in bulk.

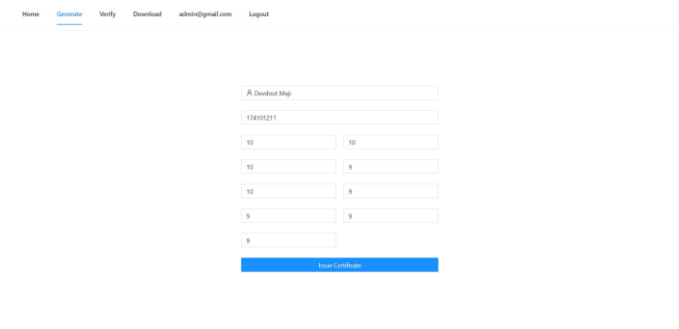


Generate Document

On Paperwork, you can generate Single Certificate or Multiple Certificate by uploading CSV.

Fig. 6. Two Options for Generation of Certificate

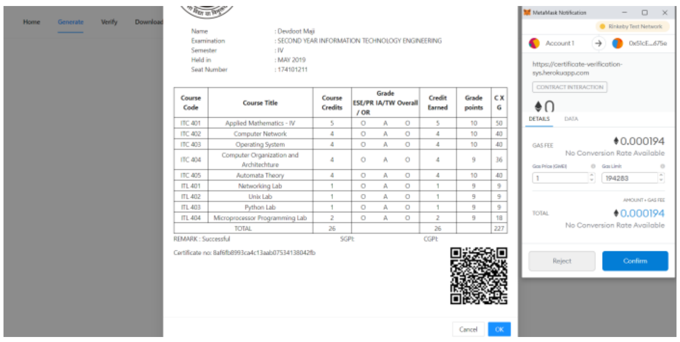
The Certificate issuer will first be asked to feed in their certificate template. Depending upon the number of fields the issuer will be asked to fill in the data of the respective fields. Our system will grab that data and append it to the pre-inserted template.



Devdoot Maji	
174101211	
10	10
10	9
10	9
9	9
9	

Fig. 7. Filling of details for generating Certificate

A preview will be generated for the Document. If the issuer approves it the IPFS will run its hashing algorithm and store it. The same hash will be forwarded to Blockchain Node and the issuer needs to approve the Document Generation charges in Metamask and it will be saved in the Blockchain.



Name : Devdoot Maji
Examination : SECOND YEAR INFORMATION TECHNOLOGY ENGINEERING
Semester : IV
Held in : MAY 2019
Seat Number : 174101211

Course Code	Course Title	Course Credits	Grade ESE/PR / OR	IA/TW	Overall	Credit Earned	Grade points	C X G
ITC 401	Applied Mathematics - IV	5	O	A	O	5	10	50
ITC 402	Computer Network	4	O	A	O	4	10	40
ITC 403	Operating System	4	O	A	O	4	10	40
ITC 404	Computer Organization and Architechture	4	O	A	O	4	9	36
ITC 405	Automata Theory	4	O	A	O	4	10	40
ITL 401	Networking Lab	1	O	A	O	1	9	9
ITL 402	Unix Lab	1	O	A	O	1	9	9
ITL 403	Python Lab	1	O	A	O	1	9	9
ITL 404	Microprocessor Programming Lab	2	O	A	O	2	9	18
TOTAL		26				26		227

REMARK : Successful SGPI: CGPI:

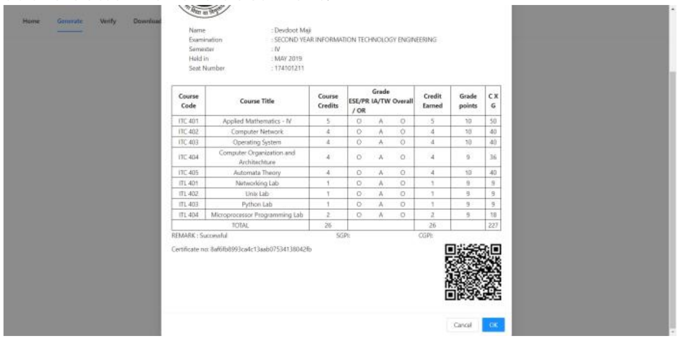
Certificate no: 8af6fb8993ca4c13aab07534138042fb

https://certificate-verification-sys.herokuapp.com

GAS FEE 0.000194 No Conversion Rate Available
Gas Price (GWEI) 1 Gas Limit 194283
AMOUNT + GAS FEE
TOTAL 0.000194 No Conversion Rate Available

Fig. 8. MetaMask Notification for Payment charges of generating Certificate

The document will be generated with its unique ID printed at the bottom of the document.



Name : Devdoot Maji
Examination : SECOND YEAR INFORMATION TECHNOLOGY ENGINEERING
Semester : IV
Held in : MAY 2019
Seat Number : 174101211

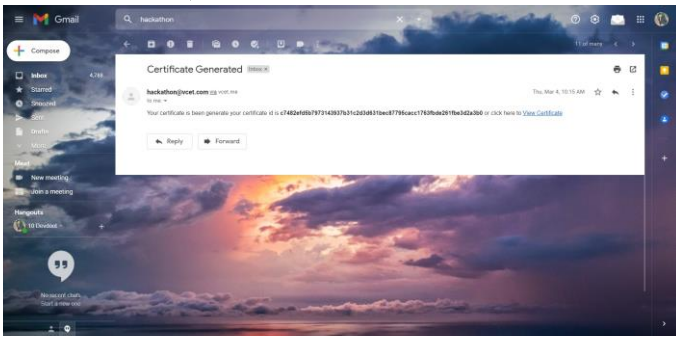
Course Code	Course Title	Course Credits	Grade ESE/PR / OR	IA/TW	Overall	Credit Earned	Grade points	C X G
ITC 401	Applied Mathematics - IV	5	O	A	O	5	10	50
ITC 402	Computer Network	4	O	A	O	4	10	40
ITC 403	Operating System	4	O	A	O	4	10	40
ITC 404	Computer Organization and Architechture	4	O	A	O	4	9	36
ITC 405	Automata Theory	4	O	A	O	4	10	40
ITL 401	Networking Lab	1	O	A	O	1	9	9
ITL 402	Unix Lab	1	O	A	O	1	9	9
ITL 403	Python Lab	1	O	A	O	1	9	9
ITL 404	Microprocessor Programming Lab	2	O	A	O	2	9	18
TOTAL		26				26		227

REMARK : Successful SGPI: CGPI:

Certificate no: 8af6fb8993ca4c13aab07534138042fb

Fig. 9. Unique ID of the generated Certificate at the bottom

After successful generation of certificate the respective student is sent a mail with their certificate id and a link to view a certificate.



Certificate Generated Inbox

hackathon@vcet.com via vcet.me
to me
Thu, Mar 4, 10:15 AM

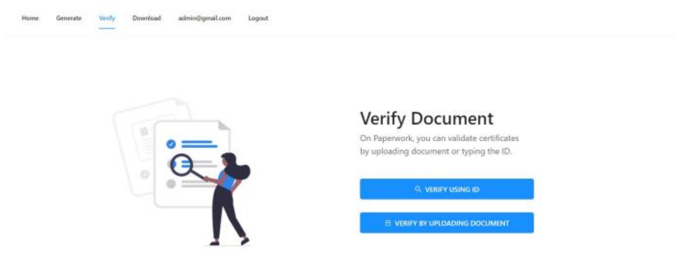
Your certificate is been generate your certificate id is c7482efd5b7973143937b31c2d3d631bec87795cacc1763fbde261fbe3d2a3b0 or click here to View Certificate

Fig 10. Mail sent to a student

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 03, March-2021

C. Validator-

The Validator is the organization's authority who will need to validate the originality of the document when a student comes for admission. He will have two options – Validate using uploading document, Validate by entering ID.



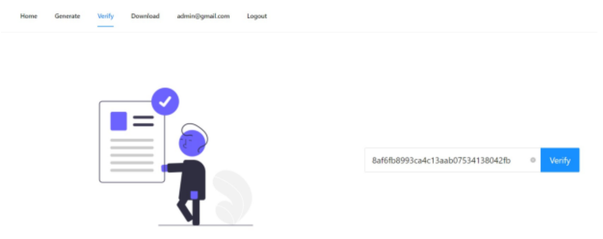
Verify Document

On Paperwork, you can validate certificates by uploading document or typing the ID.

Fig. 11. Two Options for Verification

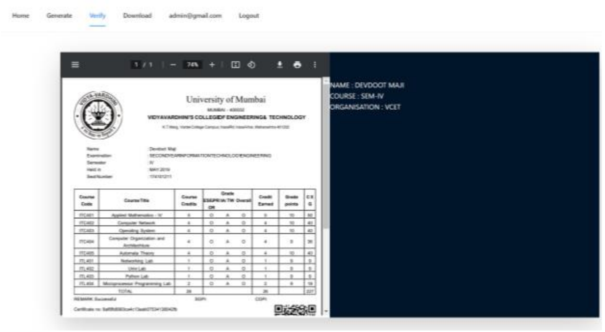
1) Validate using ID-

The authority can insert or type in the unique ID for the document which he wants to validate. If the ID matches the ID stored in the BLockchain a preview of the Document is generated along with basic details. If a fake ID is given an error is raised.



8af6fb8993ca4c13aab07534138042fb

Fig. 12. ID input being given for validation



NAME : DEVDOOT MAJI
COURSE : SEM-IV
ORGANISATION : VCET

University of Mumbai
MUMBAI - 400032
VIDYAVARDHINI'S COLLEGE OF ENGINEERING & TECHNOLOGY
K.T.Marg, Vartak College Campus, Vasai Rd, Vasai-Virar, Maharashtra 401202

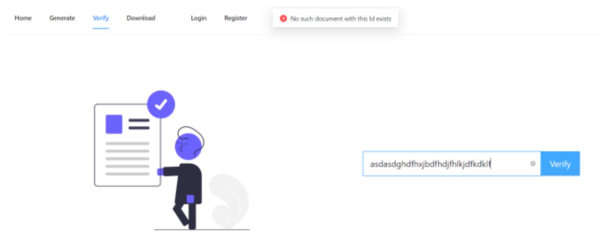
Name	: Devdoot Maji
Examination	: SECONDYEARINFORMATIONTECHNOLOGYENGINEERING
Semester	: IV
Held in	: MAY 2019
Seat Number	: 174101211

Course Code	Course Title	Course Credits	Grade ESE/PR/OR	Grade IA/TW	Grade Overall	Credit Earned	Grade points	C X G
ITC401	Applied Mathematics - IV	5	O	A	O	5	10	50
ITC402	Computer Network	4	O	A	O	4	10	40
ITC403	Operating System	4	O	A	O	4	10	40
ITC404	Computer Organization and Architechture	4	O	A	O	4	9	36
ITC405	Automata Theory	4	O	A	O	4	10	40
ITL401	Networking Lab	1	O	A	O	1	9	9
ITL402	Unix Lab	1	O	A	O	1	9	9
ITL403	Python Lab	1	O	A	O	1	9	9
ITL404	Microprocessor Programming Lab	2	O	A	O	2	9	18
	TOTAL	26				26		227

REMARK: Successful SGPI: CGPI:

Certificate no: 8af6fb8993ca4c13aab07534138042fb

Fig. 13. Successful Verification of Document generates preview and prints the data



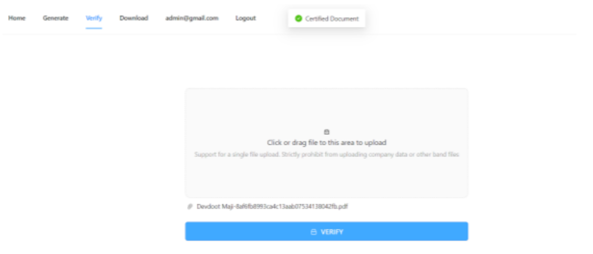
No such document with this Id exists

asdasdghdfhxjbdfhdjfhlkjdfkdklf

Fig. 14. An invalid Certificate ID being rejected

2) Validate by Uploading Document-

Another option for the validator is to directly upload the document to the system. The document's data is hashed and matched with the hash stored in IPFS and depending on the matching output is forwarded i.e. Valid or Invalid.



Certified Document

Click or drag file to this area to upload

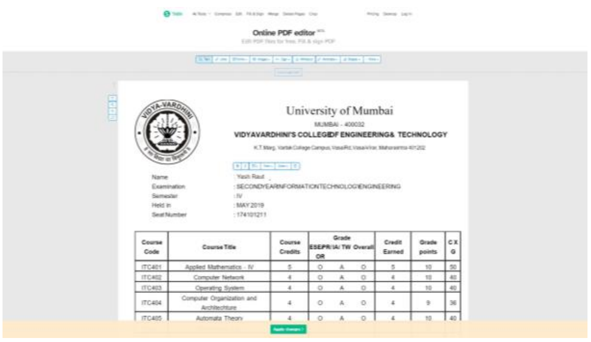
Support for a single file upload. Strictly prohibit from uploading company data or other band files

Devdoot Maji-8af6fb8993ca4c13aab07534138042fb.pdf

Fig. 15. Successful Verification of Document by Uploading the Document

3) Tampering the Document-

Now let us try to do a live example. We are tampering with the document generated by the system. The name 'Devdoot Maji' is changed to 'Yash Raut' by illegal means. Now the system tries to match the document and it classifies it as invalid. This solves the problem of fake documents.

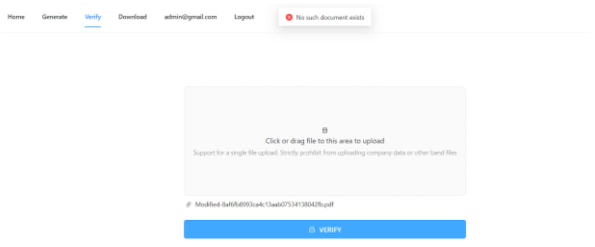


University of Mumbai
MUMBAI - 400032
VIDYAVARDHINI'S COLLEGE OF ENGINEERING & TECHNOLOGY
K.T.Marg, Vartak College Campus, Vasai Rd, Vasai-Virar, Maharashtra 401202

Name	: Yash Raut
Examination	: SECONDYEARINFORMATIONTECHNOLOGYENGINEERING
Semester	: IV
Held in	: MAY 2019
Seat Number	: 174101211

Course Code	Course Title	Course Credits	Grade ESE/PR/OR	Grade IA/TW	Grade Overall	Credit Earned	Grade points	C X G
ITC401	Applied Mathematics - IV	5	O	A	O	5	10	50
ITC402	Computer Network	4	O	A	O	4	10	40
ITC403	Operating System	4	O	A	O	4	10	40
ITC404	Computer Organization and Architechture	4	O	A	O	4	9	36
ITC405	Automata Theory	4	O	A	O	4	10	40

Fig. 16. Student Name is changed from 'Devdoot Maji' to 'Yash Raut' through illegal means



No such document exists

Click or drag file to this area to upload

Support for a single file upload. Strictly prohibit from uploading company data or other band files

Modified-8af6fb8993ca4c13aab075341380424fb.pdf

Fig. 17. Tampered Document being Rejected by the System

VI. CONCLUSION

Creating immutable ledgers is one of the main values of Blockchain. This behavior helps us to achieve a system in which all the process is transparent and unchangeable. Our System automates the process of generating Certificates and reduces the manual work needed for the verification of the same. Students are also at comparatively low risk of losing the certificate. By using an additional hashing algorithm we are decreasing the percentage of data being tampered with. The Hash of the certificate is being stored in the blockchain while the original document will be stored in the Inter Planetary File System (IPFS). This will help us preserve the data and create transparency.

ACKNOWLEDGMENT

The author would like to thank our guide Prof. Bharati Gondhalekar and all the staff of the Information Technology Department for their constant support and encouragement they gave us. We are very thankful for all the faith they had in us. And last but not least, our friends and family for their support. The dedication of our team members and enthusiasm helped us to move forward.

REFERENCES

[1] Zibin Zheng , Shaoan Xie, Hong-Ning Dai, Xiangping Chen , " An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends", IEEE 6th International Congress on Big Data, 2017.

[2] Jiin-Chiou, Narn-Yih Lee, Chien Chi, YI-Hua Chen, "Blockchain and Smart Contract for Digital Certificate," Proceedings of IEEE International Conference on Applied System Innovation 2018.

[3] Maharshi Shah, Priyanka Kumar, "Tamper Proof Birth Certificate Using Blockchain Technology", International Journal of Recent Technology and Engineering (IJRTE), Volume-7, Issue-5S3, February 2019.

[4] Emmanuel Nyaletey, Reza M. Parizi, Qi Zhang, Kim-Kwang Raymond Choo, "BlockIPFS - Blockchain-enabled Interplanetary File System for Forensic and Trusted Data Traceability", IEEE International Conference on Blockchain, 2019.

[5] Gunit Malik, Kshitij Parasrampuria, Sai Prasanth Reddy, Dr. Seema Shah, "Blockchain Based Identity Verification Model", International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019.