**A**

**Mini Project Report**

**On**

“BLOCKCERTIFY: A SECURE VERIFICATION AND VALIDATION OF EDUCATIONAL CERTIFICATES USING BLOCKCHAIN TECHNOLOGY”

Submitted in partial fulfillment of the

Requirements for the award of the degree of

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

**By**

**Gowlikar Ajitesh – 20R21A05F1**

**M. Hitesh – 20R21A05F9**

**Shaik Ahmed – 20R21A05H2**

**K. Abhilash – 21R25A0517**

Under the guidance of

**Mrs. Ragini Patil**

**Associate Professor**

**Department of Computer Science & Engineering**

**2022**



**Department of Computer Science & Engineering**

**CERTIFICATE**

This is to certify that the project entitled **“BlockCertify: A Secure Verification and Validation of Educational Certificates using Blockchain Technology”** has been submitted by **Vishal Dhatrika (18R21A05C0), Dachiraju Hemanth Kumar (18R21A0571), K Harish Chandra Prasad (18R21A0581) and Sanagala Praveen (18R21A05B0)** in partial fulfillment of the requirements for the award ofdegree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**Internal Guide** **Head of the Department**

**External Examiner**

**i**



**Department of Computer Science & Engineering**

**DECLARATION**

We hereby declare that the project entitled **“BlockCertify: A Secure Verification and Validation of Educational Certificates using Blockchain Technology”** is the work done during the periodfrom **August 2022 to December 2022** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

**Vishal Dhatrika 18R21A05C0**

**Dachiraju Hemanth Kumar 18R21A0571**

**K Harish Chandra Prasad 18R21A0581**

**Sanagala Praveen 18R21A05B0**

**ii**



**Department of Computer Science & Engineering**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our guidance for all of them.

First of all, we would like to express our deep gratitude towards our internal guide **Dr. P. CHINNASAMY, Associate Professor, Department of CSE** for his support in the completion ofour dissertation. We wish to express our sincere thanks to **Dr. E. ANUPRIYA,** HOD, Dept. of CSE and also principal **Dr. K. SRINIVAS RAO** for providing the facilities tocomplete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**Vishal Dhatrika 18R21A05C0**

**Dachiraju Hemanth Kumar 18R21A0571**

**K Harish Chandra Prasad 18R21A0581**

**Sanagala Praveen 18R21A05B0**

**iii**



**Department of Computer Science & Engineering**

**ABSTRACT**

Verifying documents authenticity is a tedious task, as involves a lot of communication mechanisms and protocols. For a certificate, or document to be verified after it is generated, the third party (or) person needs to contact the issuing body and verify its authenticity. This is a cumbersome task as it may involve multiple issuing organizations and stakeholders. We propose a blockchain based application or framework enter transactions in the blockchain network involving metadata regarding the generated document. When an issuing organization generates the original document (ex: JNTUH generating transcript), there must be a transaction on the blockchain recorded with data pertaining to the document generated (ex: hash value of the file after encryption). This framework requires the verifiers to be aware of the user credentials of the generator (ex: hiring company needs to know the unique wallet ID, or user ID in the blockchain to differentiate the real generator from a malicious user or scammer). When the verifier wants to verify if a document is legitimate or not, they can upload it to the application, where the application calculates the hash of submitted document and crosschecks it with the existing data stored in the blockchain. If the hash values of initial document and the submitted document matches, it means that the file is legitimate and has not been modified in any sort. If even any small change is made to the file, and is proposed for verifying, the hash values do not match with that in the blockchain transactions and hence deems the file as modified and un-authentic. Since the blockchain is a secure chain of individual blocks of transactions (like a ledger/ log book) there cannot be a chance of hacking or modification of the blocks after it is registered once. This mechanism involves development of a software application that performs all the aforementioned tasks by interacting with the blockchain.

**iv**

**LIST OF FIGURES & TABLES**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Name of the Figure** | **Page Number** |
| 1 | Workflow of the Architecture explaining the process of Creation of an NFT Certificate | 13 |
| 2 | Workflow of the Architecture explaining the process of Verification of an NFT Certificate | 15 |
| 3 | Connecting MetaMask wallet to BlockCertify Application | 42 |
| 4 | Selecting one account in MetaMask | 43 |
| 5 | Giving the App permission to make transactions from Account (wallet) | 43 |
| 6 | Prompt display that Wallet has been connected | 44 |
| 7 | Upon inputting recipient wallet and file location, transaction estimated fee is shown in MetaMask | 44 |
| 8 | Upon confirming the fee, transaction is Initiated and Transaction ID is shown. | 45 |
| 9 | The transaction can be verified using Transaction ID obtained. Transaction Verification can also additionally be done with file, and file hash. | 45 |

|  |  |  |
| --- | --- | --- |
| **Table Number** | **Name of the Table** | **Page Number** |
| 1 | Possible combinations of Certifier and Certificates, with their result by using this application | 17 |
| 2 | Transaction time and costs in Ropsten Ethereum Network for NFT Generation | 42 |

**v**

**INDEX**

**Certificate**  i

**Declaration**  ii

**Acknowledgement** iii

**Abstract** iv

**List of Figures and Tables**  v

**Chapter 1**

**Introduction**  1

* 1. Overview 1

1.2 Purpose of the project 1

1.3 Motivation 2

**Chapter 2**

**Literature Survey** 3

2.1 Existing System 3

2.2 Disadvantages of Existing System 6

**Chapter 3**

**Proposed System**  7

3.1 Proposed System 7

3.2 Advantages of Proposed System 7

3.3 System Requirements 8

3.3.1 Software Requirements 8

3.3.2 Hardware Requirements 8

3.3.3 Implementation Technologies 9

**Chapter 4**

**System Design** 11

4.1 Proposed System Architecture

4.2 Application Modules 11

4.2.1 Create NFT Certificate Module 11

4.2.2 Verify NFT Certificate Module 14

**Chapter 5**

**Implementation** 17

5.1 Implementation with Hypothetical Scenarios 17

5.1.1 A legitimate certificate, Legitimate Certifier 17

5.1.2 An illegitimate certificate, Legitimate Certifier 18

5.1.3 A certificate, an illegitimate Certifier 19

5.2 Source Code 21

index.html 21

styles.css 24

loading.css 26

loading.js 26

typingeffect.js 27

verify.html 27

verify.css 29

verify.js 31

create.html 36

create.css 38

create.js 39

BlockCertify.sol 41

**Chapter 6**

**Results** 42

**Chapter 7**

**Conclusion** 46

**Future Enhancements and Discussions** 46

**References** 47

# CHAPTER 1

# INTRODUCTION

* 1. **OVERVIEW**

Educational documents or any certificates are meant to be a standing proof of the purpose they are created for. A degree or a diploma certificate is used to prove a person’s abilities and educational qualifications. At the same time, it is very important for people to verify the authenticity of these documents. When fake documents circulate in the society, there is no distinguishable identification of such fake documents from legitimate ones. The verification mechanisms used today are very old, time consuming and for most part is not universally applicable. We propose an open-to-use application that can be used by Educational Institutions, Students and Verifiers (Employers) to generate NFT based certificates against existing and new educational documents that serve as a digital certificate that can be verified remotely and easily. This makes the whole process of the credit evaluation and document verification very simple and hassle-free.

* 1. **Purpose of the project**

The purpose of this project is to develop a cross-platform web application that is capable of generation and verification of educational documents. This application aims to make the experience of certificate generation and verification very easy and fool-proof. Whenever an employer or an institution wants to verify a candidate’s credentials and scores in their degrees or certificates, they can simply use this application to verify the file given by the student, and find out whether it is legitimate.

* 1. **Motivation**

Certificate Frauds are very rampant even in today’s day and age, despite having very advanced technology. Holograms, Embossed papers, and a lot of anti-fraud techniques are being broken, and it is high-time that the technology makes the process of certificate verification simple and trustworthy. There have also been instances of US Embassies encountering fake degrees and certificates, which they couldn’t verify on the background due to lack of proper communication channels between Education Departments, and Institutions. Imagine the scale of the problem, when globalization comes into play and credential evaluation are a proper business to some conglomerates. All these situations deeply intrigued us to develop a breakthrough application with the Blockchain Technology to overcome this problem.

**CHAPTER 2**

**LITERATURE SURVEY**

An extensive literature survey has been conducted by studying existing systems of Certificate verification and generation. A good number of research papers, journals, and publications have also been referred before formulating this survey.

* 1. **Existing System**

The existing system of Certificate Generation and Validation involves the Certifying Authority generating either Hard-Copied Certificates with some security and verification mechanisms like tamper proof holograms, embossed papers, etc. Also, there are digitally signed certificates where the digital signature on a file can be verified. But for the digital signatures to be verified, it is not very user-friendly.

The responses to various research articles are documented below by the order of the number that have been used to specify them in the references in the end.

The Geetha et al [1] mainly focuses on user registration-based certificate generation which has been achieved with the help of AES Encryption algorithm. The risk involved here is that any third person can fraudulently register as a Certifying organization.

The Priya et al [2] emphasizes on E-certificates generated on the app and let the users verify them anywhere. The limitation to this approach is that old physical documents cannot be secured, as the data inputs are only used to automatically generate an e-certificate. This method was implemented with the help of an Ethereum private simulated blockchain, and Solidity smart contracts.

The Lamkoti et al [3] also focuses on students and organizations to register on the portal to verify and access certificates on the blockchain. This has been achieved by using the technologies, IPFS, Ganache to simulate a private blockchain network. Pitfall here is the usage of a private blockchain which can be manipulated.

The shah et al [4] discusses how birth certificates can be stored online and used to verify on the blockchain based on permission-based access, making use of IPFS and cryptographic file hash techniques.

The Gayathiri et al [5] discussed digital certificate generation and validation but included no technological stack use or methodology. Mentioned that the chaotic algorithm is used to generate the hash code value for the certificate. Then the certificates are stored in the blockchain. And these certificates are validated by using the mobile application

The Rojiyati et al [6] discussed land records management by the virtue of blockchain technology, performing land transactions and maintaining digital ledger for properties highly eliminating chance or fake documents or illegal encroachments.

In Gopal et al [7], emphasizes the idea of E-certificate generation for the records in blockchain, limited by viewing rights of certificate, only verified and registered personnel can verify certificates.

The Baldi et al [8] discussed certificate validation through public ledgers involving multiple Cas and by verifying the time period. The limitation to this method was that the certificates were time-bound and carried an expiry date.

The Wang et al [9] explored certificate transparency and revocation transparency using the tools Firefox and Nginx. It identifies the fraud sign certificates, revocation status information, limitation is it accepts it only if it is published and not revoked.

In Nitin et al [10] observed that they create Smart contracts used to create certificates or to either verify & hash details add to the blockchain but this application is only supported for educational institutions, the network verifies and saves the same data in the application

The Trong et al [11] is mentioned that it Implement an issuing and verifying model called UniCert based on UniCoin they did not mention any particular technology used but observed that it is Based on private or limited blockchain making it hard to implement or pay for.

The Jiin et al [12] explores that the app generates e-certificates with a limited blockchain technology limitation is they generate QR code which is mandatory to check or verify the certificate through any device.

The Zyskind et al [13] discussed how blockchain can not only be used for financial transactions, but data transfer and privacy management tasks, by proposing a protocol to manage data privacy in a distributed way with blockchain.

The Haidar et al [14] explores the possibility of examination conducting, transcripts and certificate generation, all in a single software using blockchain technology, which when implemented can be a deal-breaker in the way we conduct exams today. But this idea only explored the possible uses, without specifying any technological methodology to achieve it.

The Gopal et al [15] Generates e-certificates receiver verify by digital sign stores in blockchain and generated serial number is made use of QR codes to verify the authenticity of certificates generated and issued on a private blockchain. Used to verify/QR (mandatory), students use it here e-certificates are required proven by digital signature.

* 1. **Disadvantages of Existing system**:

Concisely summarizing the disadvantages of the above implementations:

* Use of private blockchain prone to accepting wrong or illegal transactions.
* Requirement of Login to application limits the scope of verification.
* Applications automatically generating certificates based on inputs could not certify already existing hard-copy documents.
* Risk of any third party fraudulently registering as an Certifying Authority.
* Use of hard copy documents prone to fakes.
* Hardcopy documents prone to attack or damage by natural or man-made disasters.
* Sorting and organizing hard copied documents is not effective.
* Loss of document incurs cost and effort to obtain a replacement.

**CHAPTER 3**

**PROPOSED SYSTEM**

**3.1 Proposed System**

The proposed approach is to develop a cross-platform web application that runs on a server and performs the blockchain transactions pertaining to certificate generation on the Ethereum Blockchain. For demonstration purposes, The Ropsten Test Network has been utilized. The proposed system has some prerequisites that need to be fulfilled before the whole workflow can be assumed to be complete. Initially, the Certifying Institutions, or Authorities, like Schools and Colleges need to have their Ethereum Account Address (Public Wallet Address/ID) declared publicly on their websites for the verifiers to verify. This has to be specified on their respective website.

**3.2 Advantages of Proposed System**

The proposed system has the following advantages:

* Globally verifiable certificates/documents
* Accessed with a Transaction ID only
* Blockchain Transaction ensures data can not be modified once committed.
* Secure documents storage on Institution Servers.
* Close-to-zero time delay for verification of certificates.
* Effective Identification of Modified/Photoshopped documents.
* Easy transfer of documents from person to person in the form of NFTs.

**3.3 SYSTEM REQUIREMENTS**

The system requirements for the development and deployment of the project as an application are specified in this section. These requirements are not be confused with the end-user system requirements. There are no specific, end-user requirements as the intended application is cross-platform and is supposed to work on devices of all form-factors and configurations.

**3.3.1 SOFTWARE REQUIREMENTS**

Below are the software requirements for application development:

1. Editor for HTML, CSS and JavaScript- VS Code, or Notepad++
2. Editor for Solidity: Remix IDE on Chrome Browser
3. Google Chrome, Firefox, Microsoft Edge or Brave Browser with Extension Support.
4. MetaMask extension installed in any of the above browser.

**3.3.2 HARDWARE REQUIREMENTS**

Hardware requirements for application development are as follows:

* 1. CPU- intel i3 or higher
  2. RAM – 4 GB or higher

Apart from hardware and software requirements, the certifying authority needs to have enough Ethereum (ETH) Cryptocurrency in their linked MetaMask wallet to pay as transaction fees.

**3.3.3 IMPLEMENTATION TECHNOLOGIES**

**Blockchain:**

A block-chain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. As blocks each contain information about the block previous to it, they form a chain, with each additional block reinforcing the ones before it. Therefore, block-chains are resistant to modification of their data because once recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks.

**Ethereum Blockchain:**

Ethereum is a decentralized, open-source blockchain that supports smart contracts. Ether is the platform's native cryptocurrency. Anyone may develop permanent and irreversible decentralized apps on Ethereum, with which users can interact. Decentralized finance (DeFi) apps provide a wide range of financial services without the need for traditional financial intermediaries such as brokerages, banks, or exchanges, such as allowing cryptocurrency users to borrow against or lend out their holdings for interest. Users may also produce and exchange NFTs on Ethereum.

**Smart Contracts:**

To expedite transactions, a set of rules known as a Smart Contract is placed on the blockchain and automatically executed upon function calls. For example, a smart contract can specify the terms for examination, smart cities, corporate bond transfers, or terms for travel insurance payments, NFT creation, etc.

Ethereum has developer-friendly languages for writing smart contracts:

* Solidity
* Vyper

Smart contracts are public on Ethereum and can be thought of as open APIs. That means you can call other smart contracts in your own smart contract to greatly extend what's possible. Contracts can even deploy other contracts. We have made use of Remix IDE to compile and deploy our smart contract.

**NFT:**

NFT is an abbreviation for non-fungible token. Non-fungible is an economic word that can be used to illustrate items such as your furniture, a song file, or your computer. Because of their distinct features, some goods cannot be substituted with others. NFTs are tokens that may be used to indicate ownership of one-of-a-kind goods. They allowed us to tokenize items such as art, valuables, and even real estate. NFTs are digitally distinct; no two NFTs are alike. This very concept can be utilized to create NFTs of physical or digital documents and then these tokens can be used to identify the real document from a counterfeit.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Proposed System ARCHITECTURE**

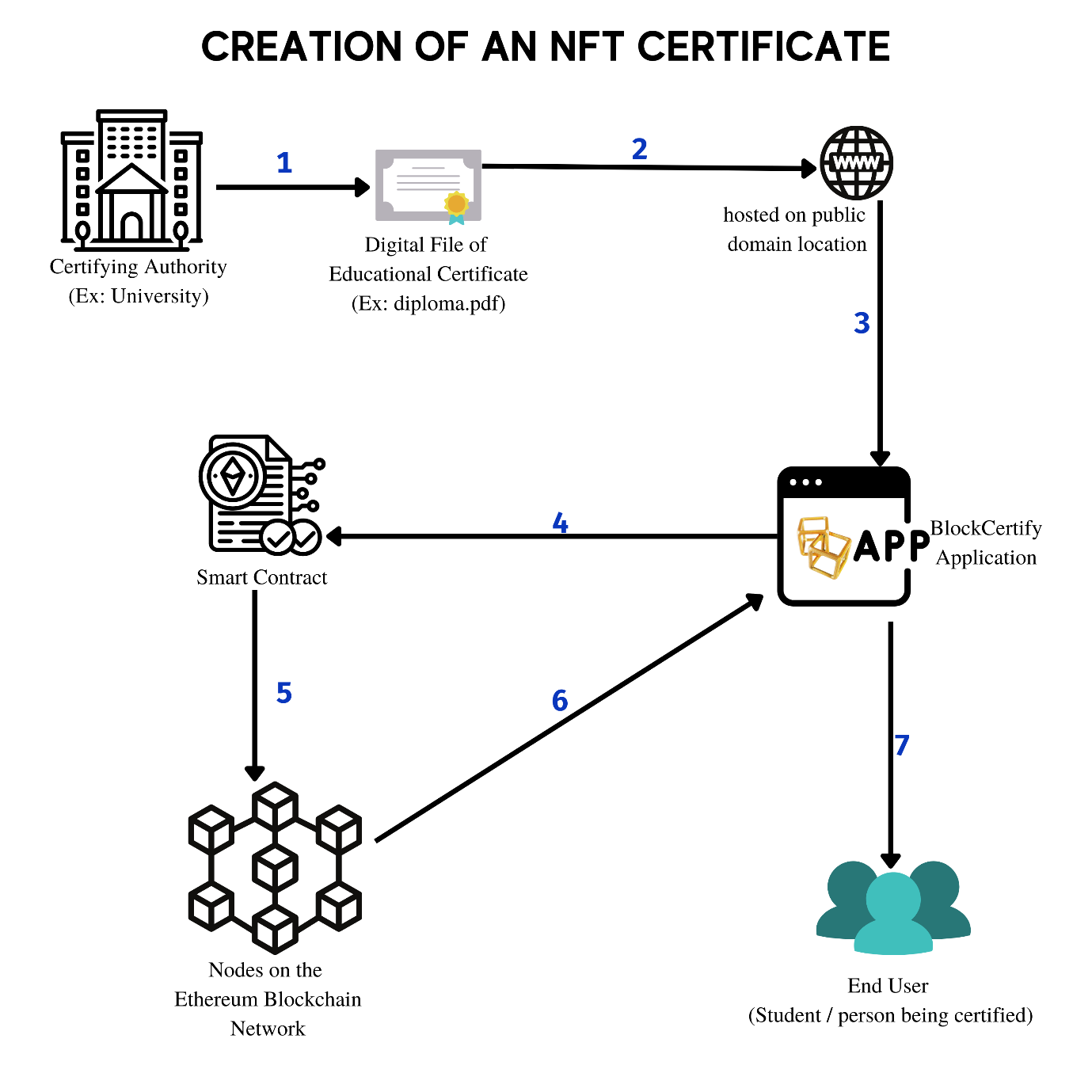
The proposed system involves the development of a web application that can be used on any platforms and configurations of devices. The application has been named BlockCertify, and this refers to the application developed hereafter.

**4.2 APPLICATION MODULES**

The application on an overall involves two main modules, which cater to the two main functions of this application, i.e., to verify certificate and to generate new certificates.

**4.2.1 Create NFT Certificate Module:**

This module involves the logic and programming code that fulfils the task of creating an NFT certificate, on the blockchain, as a form of a transaction to the receiver’s blockchain address (also referred to as the receiver’s wallet). The workflow requires the Certifying Authority to be in possession of the digital form of the certificate being made into an NFT. These forms generally include files with extensions, like, .jpg, .jpeg, or .pdf, .png, etc. When a digital file of the certificate is being used to create an NFT, since, there is no physical or hard-paper file, there can only be one legitimate file. But, when a hard-paper (hard-copy) file or document is being used to create an NFT, then only a single digital instance(scan) of the original hard copied file will be considered as original on the blockchain network. Any other scans, or photographs of the hard copied document, xerox or photocopies would not account as an original or legitimate file. There are some elements in this architecture, that play a vital role in the creation of NFT certificates with our application. The elements listed in the order of their appearance are as follows: Certifying Authority, Digital Certificate (File being made into NFT), Location of the file on the public domain, BlockCertify Application, Smart Contract, Ethereum Node (in the Ethereum Blockchain Network), End User (Student or person being certified). It is to be fundamentally understood that for anyone to make a transaction on the Ethereum Blockchain network, they must be in possession of sufficient Ether (ETH) currency to pay as a gas fee on the Ethereum network. Since BlockCertify is also based on the Ethereum Blockchain, the certifying authority needs to have sufficient Ether in their Blockchain Account (also called as wallet). We have tested our application with the MetaMask wallet, and for best results, it is highly recommended that the Certifying Authority and the End User have their accounts (wallets) set-up on MetaMask. Also, for the Certifying Authority to be able to validate requests of certificate verification, it is mandatory for the Certifying Authority to use only a limited Ethereum accounts (wallets) whose public addresses are publicly disclosed and advertised for on their domains and publications, solely for the purpose of verification. These addresses will act as an identifier to distinguish the real and legitimate Certifying Authority from an illegitimate or a fake one.

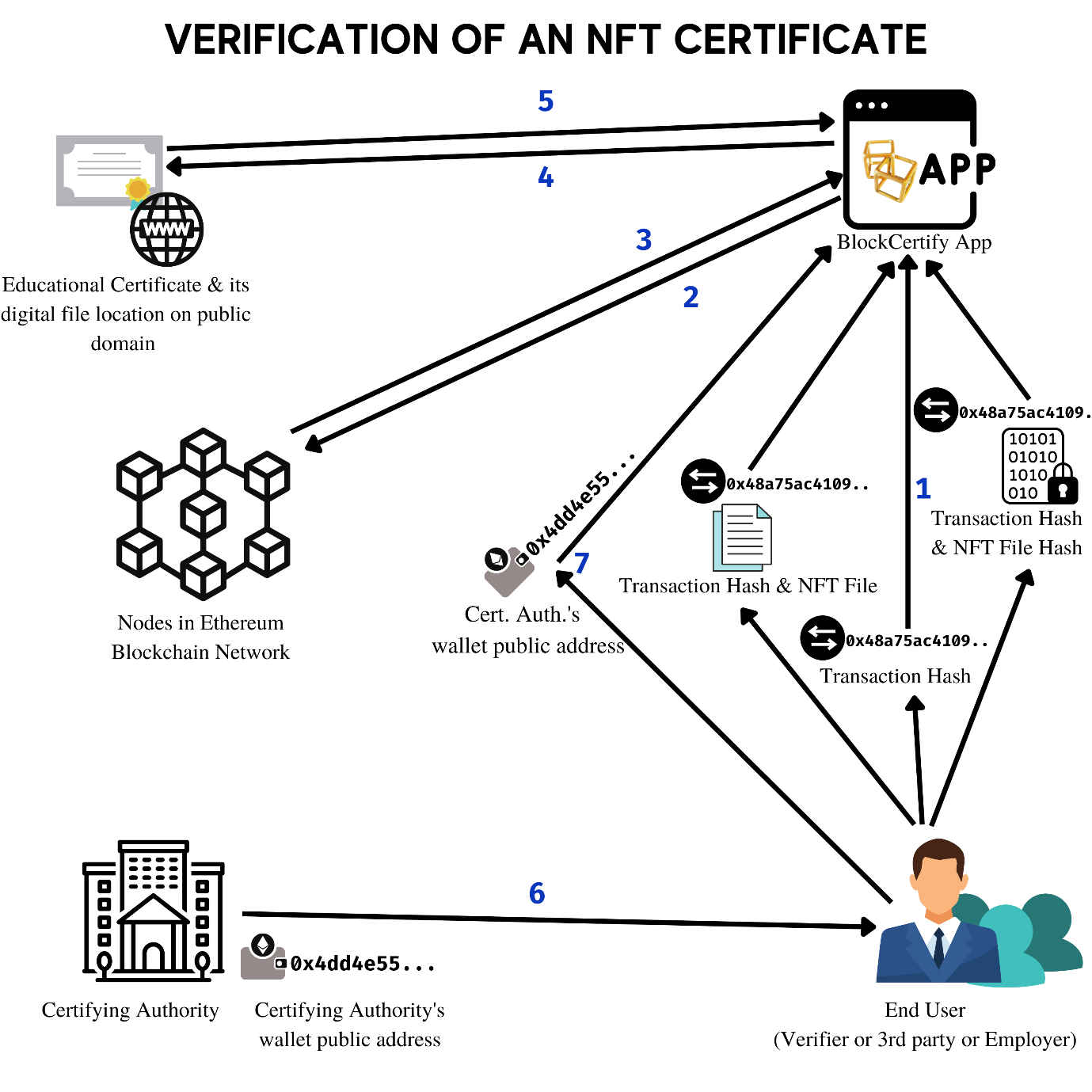


*Figure 1: Workflow of the Architecture explaining the process of Creation of an NFT Certificate*

The workflow mainly involves the following steps, starting from the Certifying Authority creating a digital certificate in the form of a file with .jpeg, .jpg, .pdf, or a .png file extension. This file will be referred to as NFT Certificate File. This particular file is hosted on public domain with a permanent URL, that does not change with time, as the certificate can be verified only by this means. For NFTs, generally IPFS file system is used to generate permanent file links, to files stored on a distributed network, but, any other stable and public file location on the internet can be used. (Ex: University file hosting on its own servers with a static IP or domain name). This location will be the location of the file on the public domain. When the certifying authority enters the Create certificate module on the BlockCertify application, they are supposed to be accessing it from a browser where their MetaMask wallet has been linked to, or from a MetaMask mobile applications’ inbuilt browser for linking the MetaMask wallet. The BlockCertify Application takes the user’s inputs namely the NFT file and the recipient’s (student’s) wallet address and invokes the smart contract which has already been deployed to create BlockCertify NFTs. This smart contract interacts with the Ethereum Blockchain, which in turn makes a transaction with the Ethereum Virtual Machine (which is the brain of the Ethereum Blockchain Network). When the transaction response arrives back to the BlockCertify application, if the transaction completes successfully, BlockCertify application displays the transaction data, like the Transaction Hash (also called Transaction ID) to the Certifying Authority which can be forwarded to the student /recipient for verification.

**4.2.2 Verify NFT Certificate Module:**

This module encompasses the frame works and functions that fulfil the task of verifying an already created NFT Certificate (created only using BlockCertify App) on the blockchain. This module enables any third party, or a verifier, or an employer to verify certificates on the blockchain, by providing the transaction hash (also sometimes referred as Transaction ID) primarily, and the file to be verified, or its file hash value. The elements in the order of their appearance in this architecture are as follows: End User (Verifier or 3rd party or Employer), Transaction Hash, NFT File, NFT File Hash value, Ethereum Node (in the Ethereum Blockchain Network), Certificate File at the publicly hosted location, BlockCertify Application, Certifying Authority, and its public wallet address.



*Figure 2: Workflow of the Architecture explaining the process of Verification of an NFT Certificate*

This module’s workflow mainly involves verifying data from the user’s inputs and comparing it with that from the blockchain data, to report any changes or to report any fraud. Since, transactions and data on the blockchain is permanent and immutable, this favours very much in safeguarding the data from hacking or modification. For the BlockCertify application to be able to get the data from the Ethereum Blockchain, it needs to be connected to a node on the Ethereum Blockchain. This has been achieved by the virtue of a connection to a node on the Ethereum blockchain via an API key obtained with Infura. Infura is a blockchain node infrastructure service that allows apps and developers to get data from, and broadcast transactions to, the Ethereum blockchain. Infura's network is utilized as a backend for Ethereum services and applications, including MetaMask and many others. This API key can be used to make an RPC Call to the node, and get results and read the data of blocks that node holds. This data can be used to verify transactions ID, obtain NFT metadata, and verify certificate generator address.

Upon entering this module, the user is presented with three choices to verify the certificate, with one being using the transaction ID (transaction hash), another being transaction hash along with the NFT File’s SHA 256 Hash, and the last method being transaction hash along with the NFT file upload. The module has algorithms and logic in place that can detect any minor change to the files, like editing even a single bit of data. This can be ensured by the means of comparison of the SHA-256 hash of the file stored at the metadata location, and the user-supplied file. If the SHA-256 hash values of both files match, then the file is said to be original. If the hashes do not match, it means that the user holds a hash/file that has been obtained as a result of modification/tampering of the original NFT file. The user can also find out that the transaction has been initiated from which particular Ethereum address (wallet address) and verify it with the Certifying Authority’s wallet address and the application gives user the response whether it was generated by that particular Certifying Authority or not. Hence, this also removes the risk of any third party or person fraudulently impersonating or pretending to be the Certifying Authority and generate NFT certificates.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 IMPLEMENTATION WITH HYPOTHETICAL SCENARIOS**

This subsection evaluates the use of application in various scenarios involving different hypothetical situations.

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Legitimate | Illegitimate | Result |
| 1 | Certificate:  *Degree.pdf* | None | Authentic |
| Certifier:  *ABC University* |
| 2 | Certifier:  *ABC University* | Certificate: *Degree-fk.pdf* | Not Authentic, Certificate Modified. |
| 3 | None | Certifier:  *XYZ University* | Not Authentic, Certifying Authority is not legitimate. Cannot assess certificate. |

Table 1: Possible combinations of Certifier and Certificates, with their result by using this application.

**5.1.1 A legitimate certificate, Legitimate Certifier**

1. At ABC University’s end:

This scenario involved a legitimate certificate (say Degree.pdf) has been used to create an NFT on the BlockCertify Application. This transacation was carried out by ABC University which in this case is a legitimate certifier and university. The univeristy creates the NFT using the Create Certificate module on the BlockCertify Application. First, the University links up their MetaMask wallet (whose wallet addess is publicly recognisable, like a domain name) to the BlockCertify Application. Next, the inputs to the Create Certificate operation, i.e, Student’s wallet Address, and the certificate file link (Location where Degree.pdf is hosted publicly on the Internet or IPFS-like web storage) are given in the app. As a result of the transaction, the Transaction ID is obtained and displayed to the University. The Univeristy gives the transaction ID to the student to verify their NFT Certificate.

1. At the Student’s end:

The student verifies the certificate using the Transaction ID provided by the University, and checks their certificate. The student can also use and give the transaction ID to anyone who wants to verify their certificate, for instance, the student can give the Transaction ID to a MNC, or Corporate Employer, or a higher education Institute.

1. At the verifier’s end:

The verifier can verify the certificate using the Transaction ID and the Certifying Authority’s wallet address. The transaction ID is obtained from the student. The certifying authority’s wallet address is directly obtained from the University or the University’s website, where they post it for verification purposes.

The verifier also uses the Verify Certificate module and finds out the certificate to be authentic.

**5.1.2 An illegitimate certificate, Legitimate Certifier**

1. At ABC University’s end:

Similar to the above case, in this case also, the Certifier, i.e., ABC University generates the NFT for the file, Degree.pdf and makes a transaction and gives the transaction ID to the student.

1. At the Student’s end:

The student takes the Transaction ID and obtains the certificate, modifies it and changes the file data, and call this file Degree-fk.pdf. This file is supplied by the student to the third parties, verifiers and employers to verify, along with the original transaction ID.

1. At the verifier’s end:

The verifier uses the BlockCertify App, and gives the Transaction ID and the file sent by student uploaded as the input for verification. The application compares the input file for modification with the actual NFT file, and reports that the file is indeed modified, and there has been alteration. In this case, the verifier finds out that an illegitimate and fake certificate has been supplied by the student.

**5.1.3 A certificate, an illegitimate Certifier**

1. At ABC University’s end:

The university similarly makes transaction and generates transaction for the student and gives the certificate to the student using the file Degree.pdf

1. At the Student’s end:

The student, by utilizing the BlockCertify application, himself generates an NFT against a file which holds a fake certificate. This transaction is completely carried out by the student himself and is completely different from the NFT transaction carried out by the ABC University. Moreover, the student uses this fake certificate transaction ID to supply to the employers and third party.

1. At the verifier’s end:

The verifier takes the transaction ID as input, and optionally may also take the file from the student. When the verifier checks the transaction ID and file on BlockCertify app’s verify module, it reports that the file is un-modified. But, when the verifier checks the certifying authority, i.e., ABC University’s Wallet address obtained from their website, it does not match, and reports that it is a transaction from somebody impersonating to be the Certifying Authority hence deeming it illegitimate.

**5.2 SOURCE CODE**

**index.html**

<!doctype html>

<html lang="en">

<head>

<title>BlockCertify | Create and Verify NFT Certificates</title>

<link rel="stylesheet" type="text/css" href="./css/styles.css">

<!-- icon -->

<link rel="icon" type="image/png" href="./images/logo.png">

<link rel="shortcut icon" type="image/png" href="./images/logo.png">

<script src="./js/loading.js"></script>

<link rel="stylesheet" href="./css/loading.css">

<!-- fonts -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Fira+Mono&family=Lato:ital@1&display=swap" rel="stylesheet">

<!-- bootstrap css -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"

integrity="sha384-zCbKRCUGaJDkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdKlfG/eSrtxUkn" crossorigin="anonymous">

</head>

<body onload="typeWriter(); hideLoading()">

<div id="loading"></div>

<header class="header">

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">

<img src="./images/logo.png" width="40" height="40" class="d-inline-block align-top" alt="">

BlockCertify</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="#home">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#about">About Us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#contact">Contact Us</a>

</li>

</ul>

</div>

</nav>

</header>

<div class="container">

<div class="row" id="home">

<div class="col-sm-6 banner-info">

<h2 class="firamono" id="TypingEffectText">Certify and Verify documents with the Blockchain

</h2>

<p class="big-text">BlockCertify</p>

<p class="firamono">Educational Certificates, Legal Docs or Bonds, create NFTs for existing physical

documents, and

verify

their

credibility from anywhere around the globe.</p>

<a class="btn btn-first" href="./verify.html">VERIFY</a>

<a class="btn btn-second" href="./create.html">CREATE</a>

</div>

<div class="col-sm-6 banner-image">

<img src="./images/pic1.png" class="img-responsive">

</div>

</div>

<div class="row" id="about">

<h3 class="text-center mb-3">ABOUT US</h3>

<p class="text-center firamono mb-5">We are a group of passionate and goal-driven students developing this

product /

software as a part of our

Bachelor's degree Project. We are making use of the Blockchain Technology to power our application,

where we

record transaction data and generate NFT Certificates by the inputs given by user.</p>

<br><br>

<h4 class="text-center">Our Team</h4><br>

<h6 class="col-md-1 text-center"> </h6>

<h6 class="col-md-2 text-center">

<ul>

<li>

<img class="profile-img" src="./profileimgs/sir.jpeg">

</li>

<li>Dr. P Chinnasamy</li>

<li>Associate Professor, Project Guide</li>

</ul>

</h6>

<h6 class="col-md-2 text-center">

<ul>

<li>

<img class="profile-img" src="./profileimgs/5c0.jpg">

</li>

<li>Vishal Dhatrika</li>

<li>UG Student, Dept. of CSE</li>

</ul>

</h6>

<h6 class="col-md-2 text-center">

<ul>

<li>

<img class="profile-img" src="./profileimgs/571.jpeg">

</li>

<li>Hemanth Kumar Dachiraju</li>

<li>UG Student, Dept. of CSE</li>

</ul>

</h6>

<h6 class="col-md-2 text-center">

<ul>

<li>

<img class="profile-img" src="./profileimgs/581.jpeg">

</li>

<li>K Harish Chandra Prasad</li>

<li>UG Student, Dept. of CSE</li>

</ul>

</h6>

<h6 class="col-md-2 text-center">

<ul>

<li>

<img class="profile-img" src="./profileimgs/5b0.jpeg">

</li>

<li>Sanagala Praveen</li>

<li>UG Student, Dept. of CSE</li>

</ul>

</h6>

<h6 class="col-md-1 text-center"> </h6>

</div>

</div>

<footer>

<div class="row">

<div class="col">

BlockCertify Project <img src="./images/logo.png" alt="" width="25px">

</div>

<div class="col" id="contact">

<div class="text-primary">Contact Us</div>

<ul>

<li>blockcertifyproject@gmail.com</li>

<li>MLR Institute of Technology</li>

<li>Dundigal, Hyderabad</li>

<li>500043</li>

</ul>

</div>

</div>

<div class="row">

<div class="col text-center text-muted">

Content owned and developed by Batch B-16 (2018-2022), Department of CSE, MLR Institute of Technology,

Hyderabad, IN.</div>

</div>

</footer>

<!-- typing effect js -->

<script src="./js/typingeffect.js"></script>

<!-- bootstrap jquery -->

<script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"

integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"

integrity="sha384-fQybjgWLrvvRgtW6bFlB7jaZrFsaBXjsOMm/tB9LTS58ONXgqbR9W8oWht/amnpF"

crossorigin="anonymous"></script>

</body>

</html>

**styles.css**

\* {

margin: 0;

padding: 0;

}

.navbar-style {

border-radius: unset;

box-shadow: 0px 5px rgb(204, 204, 204);

/\* text-transform: uppercase; \*/

}

.logo {

height: 48px;

padding: 2px 10px;

}

.icon-bar {

background: white;

}

li a {

color: rgb(48, 23, 23);

}

.banner-info .big-text {

font-size: 52px;

margin: 20px 0;

color: #666dff;

font-family: 'Lato', sans-serif;

}

a.btn {

margin: 30px 10px;

margin-bottom: 0;

width: 150px;

padding: 10px;

border-radius: 20px;

}

a.btn-first {

background: linear-gradient(to right, rgba(47, 47, 255, 0.725), rgba(51, 125, 253, 0.83), rgb(0, 157, 255), rgb(51, 187, 255), rgb(28, 213, 255));

color: #fff;

}

a.btn-second {

background: transparent;

border: 1px solid #666dff;

color: #333;

}

a.btn:hover {

background: linear-gradient(to right, rgba(47, 47, 255, 0.725), rgba(51, 125, 253, 0.83), rgb(0, 157, 255), rgb(51, 187, 255), rgb(28, 213, 255));

border: none;

color: #fff;

box-shadow: 5px 5px 10px #999;

transition: 0.3s;

}

.banner-info {

margin: 30px 0;

}

.banner-image {

margin: auto 0;

}

.navbar-brand {

text-decoration: none;

font-family: 'Lato', sans-serif;

font-size: x-large;

font-weight: bold;

vertical-align: middle;

line-height: inherit;

color: black;

}

.navbar-brand:hover {

text-decoration: none;

color: rgb(64, 64, 64);

}

.nav-item {

text-transform: uppercase;

font-weight: 600;

margin: 0.4rem;

padding: 0.3rem;

border-radius: 0.4rem;

}

.nav-item:hover {

background: rgba(217, 217, 217, 0.645);

}

.navbar {

background: linear-gradient(to right, rgba(47, 47, 255, 0.725), rgba(51, 125, 253, 0.83), rgb(0, 157, 255), rgb(51, 187, 255), rgb(28, 213, 255));

}

.col-md-2 ul li {

list-style-type: none;

margin-bottom: 10px;

}

.profile-img {

max-width: 80px;

max-height: 80px;

border-radius: 50%;

}

.img-responsive {

max-width: 100%;

vertical-align: middle;

}

.row h3,

.row h4 {

width: 100%;

}

.firamono {

font-family: 'Fira Mono', monospace;

}

footer {

max-width: 100vw;

}

footer .row {

background-color: black;

color: white;

margin-left: 0;

margin-right: 0;

}

footer .row .col {

padding: 0;

margin: 0.9rem;

}

footer li {

list-style-type: none;

}

#about {

min-height: 82vh;

margin: auto;

}

#home {

min-height: 82vh;

margin: auto;

}

**loading.css**

#loading {

top: 0;

left: 0;

width: 100%;

height: 100vh;

background: rgba(255, 255, 255, 0.81) url(https://blockcertifyproject.github.io/images/loading.gif);

position: fixed;

background-repeat: no-repeat;

background-attachment: fixed;

background-position-x: center;

background-position-y: center;

z-index: 10001;

}

**loading.js**

function hideLoading() {

document.getElementById("loading").style.display = "none";

}

**typingeffect.js**

var i = 0;

var txt = document.getElementById('TypingEffectText').innerHTML;

document.getElementById('TypingEffectText').innerHTML = '';

function typeWriter() {

var speed = 120;

if (i < txt.length) {

document.getElementById("TypingEffectText").innerHTML += txt.charAt(i);

i++;

setTimeout(typeWriter, speed);

}

}

**verify.html**

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Fira+Mono&family=Lato:ital@1&display=swap" rel="stylesheet">

<link rel="icon" type="image/png" href="./images/logo.png">

<link rel="shortcut icon" type="image/png" href="./images/logo.png">

<script src="https://cdn.ethers.io/lib/ethers-5.2.umd.min.js" type="application/javascript"></script>

<script src="./js/verify.js"></script>

<script src="./js/loading.js"></script>

<link rel="stylesheet" href="./css/verify.css">

<link rel="stylesheet" href="./css/loading.css">

<title>Verify a NFT Document</title>

</head>

<body onload="hideLoading()">

<div id="loading"></div>

<h1>Verify an NFT document generated by our site, <br>BlockCertify <img class="logo" src="./images/logo.png" alt=""

width="50px">

</h1>

<h2>

Select any one of the options below and provide required inputs

</h2>

<div class="optionlabel">

<label onclick="show1()">

<input type="radio" name="methodOfVerify" id="1" onclick="show1()">

Verify By Only Transaction ID

<br>

</label>

</div>

<form class="inputs" action="" method="post" id="form1" name="form1" style="display:none;">

<input type="text" name="mode" value="1" class="mode" readonly>

Enter Transaction ID:<br>

<input type="text" name="trxid1" id="trxid1" required>

<br><br><br><button type="button" onclick="showStatus()">Submit</button><br>

</form>

<div class="optionlabel">

<label onclick="show2()">

<input type="radio" name="methodOfVerify" id="2" onclick="show2()">

Verify By Transaction ID and File Hash Value

<br>

</label>

</div>

<form class="inputs" action="" method="post" id="form2" name="form2" style="display:none;">

<input type="text" name="mode" value="2" class="mode" readonly>

Enter Transaction ID:<br>

<input type="text" name="trxid2" id="trxid2" required>

<br><br>

Enter File Hash value:<br>

<input type="text" name="filehashvalue" id="filehashvalue" required>

<br><br><br><button type="button" onclick="showStatus()">Submit</button><br>

</form>

<div class="optionlabel">

<label onclick="show3()">

<input type="radio" name="methodOfVerify" id="3" onclick="show3()">

Verify By Transaction ID and File Upload

<br>

</label>

</div>

<form class="inputs" action="" method="post" id="form3" name="form3" style="display:none;">

<input type="text" name="mode" value="3" class="mode" readonly>

Enter Transaction ID:<br>

<input type="text" name="trxid3" id="trxid3" required><br><br>

Upload File:<br>

<input type="file" name="fileToUpload" id="fileToUpload" required>

<br><br><br><button type="button" onclick="showStatus()">Submit</button><br>

</form>

<iframe id="fileframe" src="" width="100%" style="display:none; min-height:400px;"></iframe>

<br><br>

<br>

<a href="" id="downloadbtn" target="\_blank" download style="display:none;">Download NFT File</a>

<p id="FileHashOutput"></p>

<p id="comparisionFilehash"></p>

<p id="errorMessage" style="display:none;"></p>

<br><br>

<p id="certifierWalletAddr" style="display:none;">The wallet address of certifier is : <span id="cwa"></span></p>

<br>

<input type="text" name="compareCwaIp" id="CompareCwaIp" style="display:none;">

<button id="CompareCwa" style="display:none;" type="button" onclick="compareCwa()">Compare Certifier</button>

<br><br>

<p id="cwamessage"></p>

<div class="reloadmsg" onclick="location.reload()">Check another Certificate / NFT</div>

<script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js/dist/web3.min.js"></script>

</body>

</html>

**verify.css**

body {

padding: 1% 20%;

font-family: 'Fira Mono', monospace;

background: linear-gradient(rgba(47, 47, 255, 0.725), rgba(51, 125, 253, 0.83), rgb(0, 157, 255), rgb(51, 187, 255), rgb(28, 213, 255));

background-repeat: no-repeat;

background-attachment: fixed;

}

h1,

h2 {

text-align: center;

font-family: 'Lato', sans-serif;

}

h1 {

line-height: 50px;

}

img.logo {

vertical-align: middle;

}

div.optionlabel {

margin-bottom: 14px;

}

input[type="text"] {

margin: 8px 0px;

padding: 10px 6px;

border: none;

border-radius: 4px;

width: 65%;

}

.inputs {

display: block;

margin: 20px auto;

padding: 20px;

background-color: rgba(211, 211, 211, 0.659);

border-radius: 10px;

}

.mode {

display: none;

}

button[type="button"] {

font-family: monospace;

border: none;

border-radius: 5px;

padding: 8px 15px;

background-color: rgb(229, 176, 0);

}

button[type="button"]:hover {

background-color: rgb(253, 194, 0);

}

button:disabled:hover {

background-color: rgb(229, 176, 0);

cursor: not-allowed;

}

.reloadmsg {

position: fixed;

padding: 10px;

top: 5px;

right: 5px;

z-index: 2;

background-color: rgba(255, 255, 255, 0.429);

border-radius: 10px;

}

.reloadmsg:hover {

background-color: white;

box-shadow: 10px 10px 7px rgba(70, 70, 70, 0.605);

cursor: pointer;

}

::-webkit-file-upload-button {

margin: 5px 8px;

border-radius: 5px;

padding: 10px;

border: 1px solid rgba(0, 0, 0, 0);

}

::-webkit-file-upload-button:hover {

border-radius: 5px;

padding: 10px;

border: 1px solid grey;

}

#errorMessage {

color: rgb(146, 0, 0);

border: 1px solid red;

background-color: rgba(211, 211, 211, 0.659);

border-radius: 4px;

padding: 0;

padding: 10px;

}

.successmsg {

color: rgb(0, 110, 0);

background-color: rgba(211, 211, 211, 0.659);

border: 3px solid rgb(0, 141, 0);

border-radius: 4px;

padding: 0;

padding: 10px;

}

.failuremsg {

color: rgb(110, 0, 0);

background-color: rgba(211, 211, 211, 0.659);

border: 3px solid red;

border-radius: 4px;

padding: 0;

padding: 10px;

}

**verify.js**

function buf2hex(buffer) { // buffer is an ArrayBuffer

return [...new Uint8Array(buffer)]

.map(x => x.toString(16).padStart(2, '0'))

.join('');

}

function compareCwa() {//cwa - certifier's wallet address

var c1 = document.getElementById("cwa").innerHTML.toLowerCase();

var c2 = document.getElementById("CompareCwaIp").value.toLowerCase();

var o1 = "Certifier Addresses Matched";

var o2 = "Certifier address not matched.";

if (c1.localeCompare(c2) == 0) {

document.getElementById("cwamessage").innerHTML = o1;

document.getElementById("cwamessage").className = "successmsg";

}

else {

document.getElementById("cwamessage").innerHTML = o2;

document.getElementById("cwamessage").className = "failuremsg";

}

}

function show1() {

document.getElementById("form1").style.display = "block";

document.getElementById("form2").style.display = "none";

document.getElementById("form3").style.display = "none";

}

function show2() {

document.getElementById("form1").style.display = "none";

document.getElementById("form2").style.display = "block";

document.getElementById("form3").style.display = "none";

}

function show3() {

document.getElementById("form1").style.display = "none";

document.getElementById("form2").style.display = "none";

document.getElementById("form3").style.display = "block";

}

var xhr = new XMLHttpRequest;

xhr.responseType = 'blob';

async function showStatus() {

const provider = new ethers.providers.JsonRpcProvider("https://ropsten.infura.io/v3/75c0aeb12c204ddc8a1b3c8727055a89");

let trx = "";

if (document.getElementById("1").checked == true) {

trx = document.getElementById("trxid1").value;

let flag = 0;

let res = null;

try {

res = await provider.getTransaction(trx);

document.getElementById("errorMessage").style.display = "none";

// console.log(res);

}

catch (err) {

document.getElementById("errorMessage").innerHTML = "<br>You have entered a wrong transaction ID.";

document.getElementById("errorMessage").style.display = "block";

flag = 1;

}

if (flag == 0) {

let data = Web3.utils.hexToAscii(res.data);

let fileuri = "";

if (data.includes("https://")) {

fileuri = data.slice(data.indexOf("https://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

else if (data.includes("ipfs://")) {

fileuri = data.slice(data.indexOf("ipfs://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

document.getElementById("fileframe").src = fileuri;

document.getElementById("fileframe").style.display = "block";

document.getElementById("FileHashOutput").innerHTML = await "The hash of NFT file is : ".concat(fileHashObtained);

document.getElementById("certifierWalletAddr").style.display = "block";

document.getElementById("cwa").innerHTML = res.from;

document.getElementById("CompareCwaIp").style.display = "inline";

document.getElementById("CompareCwa").style.display = "inline";

//fileuri = "https://blockcertifyproject.github.io/friendshipk.pdf";

var fileHashObtained = "";

xhr.onload = async function () {

var recoveredBlob = xhr.response;

const reader = new FileReader();

// This fires after the blob has been read/loaded.

reader.addEventListener('loadend', (e) => {

blob = new Uint8Array(e.target.result);

// fileHashObtained =

crypto.subtle.digest('SHA-256', blob).then(values => {

fileHashObtained = (buf2hex(values));

document.getElementById("FileHashOutput").innerHTML = "The hash of NFT file is : ".concat(fileHashObtained);

});

});

// calling the save method

// Start reading the blob as text.

reader.readAsArrayBuffer(recoveredBlob);

}

xhr.open('GET', fileuri);

xhr.send();

document.getElementById("downloadbtn").href = fileuri;

document.getElementById("downloadbtn").style.display = "inline";

}

}

else if (document.getElementById("2").checked == true) {

trx = document.getElementById("trxid2").value;

const filehashval = document.getElementById("filehashvalue").value.trim();

let flag = 0;

let res = null;

try {

res = await provider.getTransaction(trx);

document.getElementById("errorMessage").style.display = "none";

}

catch (err) {

document.getElementById("errorMessage").innerHTML = "<br>You have entered a wrong transaction ID.";

document.getElementById("errorMessage").style.display = "block";

flag = 1;

}

if (flag == 0) {

let data = Web3.utils.hexToAscii(res.data);

let fileuri = "";

if (data.includes("https://")) {

fileuri = data.slice(data.indexOf("https://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

else if (data.includes("ipfs://")) {

fileuri = data.slice(data.indexOf("ipfs://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

document.getElementById("fileframe").src = fileuri;

document.getElementById("fileframe").style.display = "block";

// document.getElementById("FileHashOutput").innerHTML = await "The hash of NFT file is : ".concat(fileHashObtained);

//fileuri = "https://blockcertifyproject.github.io/friendshipk.pdf";

var fileHashObtained = "";

xhr.onload = async function () {

var recoveredBlob = xhr.response;

const reader = new FileReader();

// This fires after the blob has been read/loaded.

reader.addEventListener('loadend', (e) => {

blob = new Uint8Array(e.target.result);

// fileHashObtained =

crypto.subtle.digest('SHA-256', blob).then(values => {

fileHashObtained = (buf2hex(values));

s1 = fileHashObtained.toLowerCase();

s2 = filehashval.toLowerCase();

s3 = "<br><br>Your input hash value matches with blockchain.";

s4 = "<br><br>Your input hash value DOES NOT match with blockchain. The file from which you generated a hash IS ALTERED.";

let CompareRes = "";

if (s1.localeCompare(s2) == 0) {

CompareRes = s3;

document.getElementById("comparisionFilehash").className = "successmsg";

}

else {

CompareRes = s4;

document.getElementById("comparisionFilehash").className = "failuremsg";

}

// document.getElementById("FileHashOutput").innerHTML = "The hash of NFT file is : ".concat(fileHashObtained);

document.getElementById("comparisionFilehash").innerHTML =

"You provided File hash: ".concat(filehashval).concat("<br><br>").concat("We obtained file Hash value: ").

concat(fileHashObtained).concat(CompareRes);

});

});

// calling the save method

// Start reading the blob as text.

reader.readAsArrayBuffer(recoveredBlob);

}

xhr.open('GET', fileuri);

xhr.send();

document.getElementById("downloadbtn").href = fileuri;

document.getElementById("downloadbtn").style.display = "inline";

document.getElementById("comparisionFilehash").style.display = "block";

document.getElementById("certifierWalletAddr").style.display = "block";

document.getElementById("cwa").innerHTML = res.from;

document.getElementById("CompareCwaIp").style.display = "inline";

document.getElementById("CompareCwa").style.display = "inline";

}

}

else if (document.getElementById("3").checked == true) {

var s1 = "";

trx = document.getElementById("trxid3").value;

const fileInputted = document.getElementById('fileToUpload').files[0];

let ff = new FileReader();

ff.readAsArrayBuffer(fileInputted);

ff.addEventListener('loadend', (z) => {

fileblob = new Uint8Array(z.target.result);

crypto.subtle.digest('SHA-256', fileblob).then(s1val => {

s1 = buf2hex(s1val).toLowerCase();

});

});

let flag = 0;

let res = null;

try {

res = await provider.getTransaction(trx);

document.getElementById("errorMessage").style.display = "none";

}

catch (err) {

document.getElementById("errorMessage").innerHTML = "<br>You have entered a wrong transaction ID.";

document.getElementById("errorMessage").style.display = "block";

flag = 1;

}

if (flag == 0) {

let data = Web3.utils.hexToAscii(res.data);

let fileuri = "";

if (data.includes("https://")) {

fileuri = data.slice(data.indexOf("https://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

else if (data.includes("ipfs://")) {

fileuri = data.slice(data.indexOf("ipfs://")).trim().split("").filter(char => char.codePointAt(0)).join("");

}

// document.getElementById("fileframe").src = fileuri;

// document.getElementById("fileframe").style.display = "block";

//fileuri = "https://blockcertifyproject.github.io/friendshipk.pdf";

var fileHashObtained = "";

xhr.onload = async function () {

var recoveredBlob = xhr.response;

const reader = new FileReader();

// This fires after the blob has been read/loaded.

reader.addEventListener('loadend', (e) => {

blob = new Uint8Array(e.target.result);

// fileHashObtained =

crypto.subtle.digest('SHA-256', blob).then(values => {

fileHashObtained = (buf2hex(values));

var s2 = fileHashObtained.toLowerCase();

var CompareRes = "";

if (s1.localeCompare(s2) == 0) {

CompareRes = "<br><br>Your Uploaded File and NFT File Hashes match. Your file in UN-altered.";

document.getElementById("comparisionFilehash").className = "successmsg";

}

else {

CompareRes = "<br><br>Your Uploaded File and NFT File on Blockchain Hashes DID NOT match. Your file IS altered.";

document.getElementById("comparisionFilehash").className = "failuremsg";

}

// document.getElementById("FileHashOutput").innerHTML = "The hash of NFT file is : ".concat(fileHashObtained);

document.getElementById("comparisionFilehash").innerHTML =

"You provided File hash: ".concat(s1).concat("<br><br>").concat("We obtained file Hash value: ").

concat(s2).concat(CompareRes);

});

});

// calling the save method

// Start reading the blob as text.

reader.readAsArrayBuffer(recoveredBlob);

}

xhr.open('GET', fileuri);

xhr.send();

document.getElementById("downloadbtn").href = fileuri;

document.getElementById("downloadbtn").style.display = "inline";

document.getElementById("comparisionFilehash").style.display = "block";

document.getElementById("certifierWalletAddr").style.display = "block";

document.getElementById("cwa").innerHTML = res.from;

document.getElementById("CompareCwaIp").style.display = "inline";

document.getElementById("CompareCwa").style.display = "inline";

}

}

}

**create.html**

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Fira+Mono&family=Lato:ital@1&display=swap" rel="stylesheet">

<link rel="icon" type="image/png" href="./images/logo.png">

<link rel="shortcut icon" type="image/png" href="./images/logo.png">

<script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>

<script src="./js/create.js"></script>

<script src="./js/loading.js"></script>

<link rel="stylesheet" href="./css/loading.css">

<link rel="stylesheet" href="./css/create.css">

<title>Create an NFT Document</title>

</head>

<body onload="hideLoading()">

<div id="loading"></div>

<h1>Create an NFT document on our site, <br>BlockCertify <img class="logo" src="./images/logo.png" alt=""

width="50px">

</h1>

<h2>

Connect your MetaMask wallet and provide the required inputs below to finish the NFT transaction

</h2>

<br>

<br>

<label for="1" class="inputs" style="text-align: center; vertical-align: middle;">

<button type="button" onclick="connectMetamask()">Connect MetaMask</button>

<br>

<img id="connectedImg" src="./images/tick.png" width="25rem" height="25rem" style="display:none;">

<span id="connectedMsg" style="display:none;">MetaMask Connected Successfully</span>

<p id="MetaMaskWarning"></p>

</label>

<label class="inputs" for="2">Enter Recipient's Wallet ID / Ethereum Account Address:

<br>

<br>

<input type="text" name="recpWallet" id="recpWallet" required oninput="enableButton()">

</label>

<label class="inputs" for="3">Upload the NFT File, and give its link below: (Starting with https:// or ipfs://)

<br>

<br>

<input type="text" name="filelink" id="filelink" required oninput="enableButton()">

</label>

<label for="4" class="inputs" style="text-align: center; vertical-align: middle;">

<button type="button" onclick="makeMyNFT()" id="finalSubmit" disabled>Create NFT</button><br>

</label>

<label for="5" class="inputs" style="display:none;" id="TrxResult">

Transaction Initiated !

<img src='./images/tick.gif' width="25rem" height="25rem">

<br>

<br>

Transaction ID :

<br><input type="text" id="trxidtxtbox" disabled>

<p align="center">

<img id="cpimg" src="./images/copy.png" alt="Copy" role="button" height="25px" onclick="copyTrx()">

</p>

<br>

Immediately copy this Transaction ID to a safe location.

<br><br>

Track at our Verify page:

<form action="https://blockcertifyproject.github.io/verify.html" style="display:inline">

<button type="submit">Verify</button>

</form>

</label>

</body>

</html>

**create.css**

body {

padding: 1% 20%;

font-family: 'Fira Mono', monospace;

background: linear-gradient(rgba(47, 47, 255, 0.725), rgba(51, 125, 253, 0.83), rgb(0, 157, 255), rgb(51, 187, 255), rgb(28, 213, 255));

background-repeat: no-repeat;

background-attachment: fixed;

}

h1,

h2 {

text-align: center;

font-family: 'Lato', sans-serif;

}

h1 {

line-height: 50px;

}

img.logo {

vertical-align: middle;

}

.inputs {

display: block;

margin: 20px auto;

padding: 20px;

background-color: rgba(211, 211, 211, 0.659);

border-radius: 10px;

}

input[type="text"] {

margin: 8px 0px;

padding: 10px 6px;

border: none;

border-radius: 4px;

width: 65%;

}

button[type="button"],

button[type="submit"] {

font-family: monospace;

border: none;

border-radius: 5px;

padding: 8px 15px;

background-color: rgb(229, 176, 0);

}

button[type="button"]:hover,

button[type="submit"]:hover {

background-color: rgb(253, 194, 0);

}

button:disabled:hover {

background-color: rgb(229, 176, 0);

cursor: not-allowed;

}

.res-inputs {

display: block;

margin: 20px auto;

padding: 20px;

background-color: #6ae687c4;

border-radius: 10px;

border: solid 2px #31c754;

}

#trxidtxtbox {

background: none;

border: none;

border: dotted 1px grey;

width: 95%;

}

#cpimg {

padding-left: 2px;

padding-right: 2px;

margin: 0 auto;

}

**create.js**

const contractAddr = "0x82af6F7fd6435a328ca18DCfEDD573600c3A063E";

var walletConnected = false;

async function connectMetamask() {

if (typeof window.ethereum !== 'undefined') {

const accounts = await ethereum.request({ method: 'eth\_requestAccounts' });

const account = accounts[0];

document.getElementById("connectedImg").style.display = "inline";

document.getElementById("connectedMsg").style.display = "inline";

if (walletConnected == false)

document.getElementById("connectedMsg").innerHTML += "<br>You are using account : " + account;

walletConnected = true;

}

else {

document.getElementById("MetaMaskWarning").innerHTML = "MetaMask not installed.";

}

}

function enableButton() {

if (document.getElementById("recpWallet").value.trim() != "" && document.getElementById("filelink").value.trim() != "") {

document.getElementById("finalSubmit").disabled = false;

}

else {

document.getElementById("finalSubmit").disabled = true;

}

}

const web3 = new Web3(new Web3.providers.HttpProvider('https://ropsten.infura.io/v3/75c0aeb12c204ddc8a1b3c8727055a89'));

async function makeMyNFT() {

window.contract = await new web3.eth.Contract(contractABI, contractAddr);

let receiverAddress = await document.getElementById("recpWallet").value;

let filelink = await document.getElementById("filelink").value;

const transactionParameters = {

to: contractAddr, // Required except during contract publications.

from: window.ethereum.selectedAddress, // must match user's active address.

'data': window.contract.methods.mintNFT(receiverAddress, filelink).encodeABI() //make call to NFT smart contract

};

//sign transaction via Metamask

try {

const txHash = await window.ethereum

.request({

method: 'eth\_sendTransaction',

params: [transactionParameters],

});

document.getElementById("TrxResult").style.display = "block";

document.getElementById("TrxResult").className = "res-inputs";

document.getElementById("trxidtxtbox").value = txHash;

console.log(txHash);

console.log("https://ropsten.etherscan.io/tx/" + txHash);

} catch (error) {

console.log("Trx failed");

}

}

function copyTrx() {

navigator.clipboard.writeText(document.getElementById("trxidtxtbox").value);

document.getElementById("cpimg").src = "./images/copied.png";

}

**BlockCertify.sol**

//Contract based on [https://docs.openzeppelin.com/contracts/3.x/erc721](https://docs.openzeppelin.com/contracts/3.x/erc721)

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

import "@openzeppelin/contracts/utils/Counters.sol";

import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";

contract BlockCertifyNFT is ERC721URIStorage{

    using Counters for Counters.Counter;

    Counters.Counter private \_tokenIds;

    constructor() public ERC721("BlockCertifyNFT", "NFT") {}

    function mintNFT(address recipient, string memory tokenURI)

        public

        returns (uint256)

    {

        \_tokenIds.increment();

        uint256 newItemId = \_tokenIds.current();

        \_mint(recipient, newItemId);

        \_setTokenURI(newItemId, tokenURI);

        return newItemId;

    }

}

**CHAPTER 6**

**RESULTS**

Our researchers have analysed and assessed their transactions on the Ethereum Ropsten Test Network and have presented their findings from six transactions of NFT Generation. Data of those transactions is tabulated as follows.

|  |  |
| --- | --- |
| Transaction Time | Transaction fee (in ETH) |
| 21 secs | 0.001809967652 |
| 24 secs | 0.000316305001 |
| 28 secs | 0.002058406907 |
| 126 secs | 0.038549921052 |
| 31 secs | 0.000202388581 |
| 26 secs | 0.000208694142 |

Table 2: Transaction time and costs in Ropsten Ethereum Network for NFT Generation.

The obtained results after the development of the BlockCertify application are as follows.

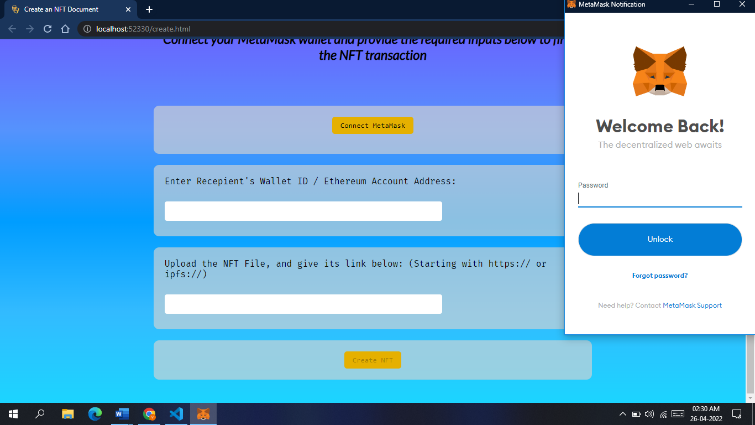


Figure 3: Connecting MetaMask wallet to BlockCertify Application

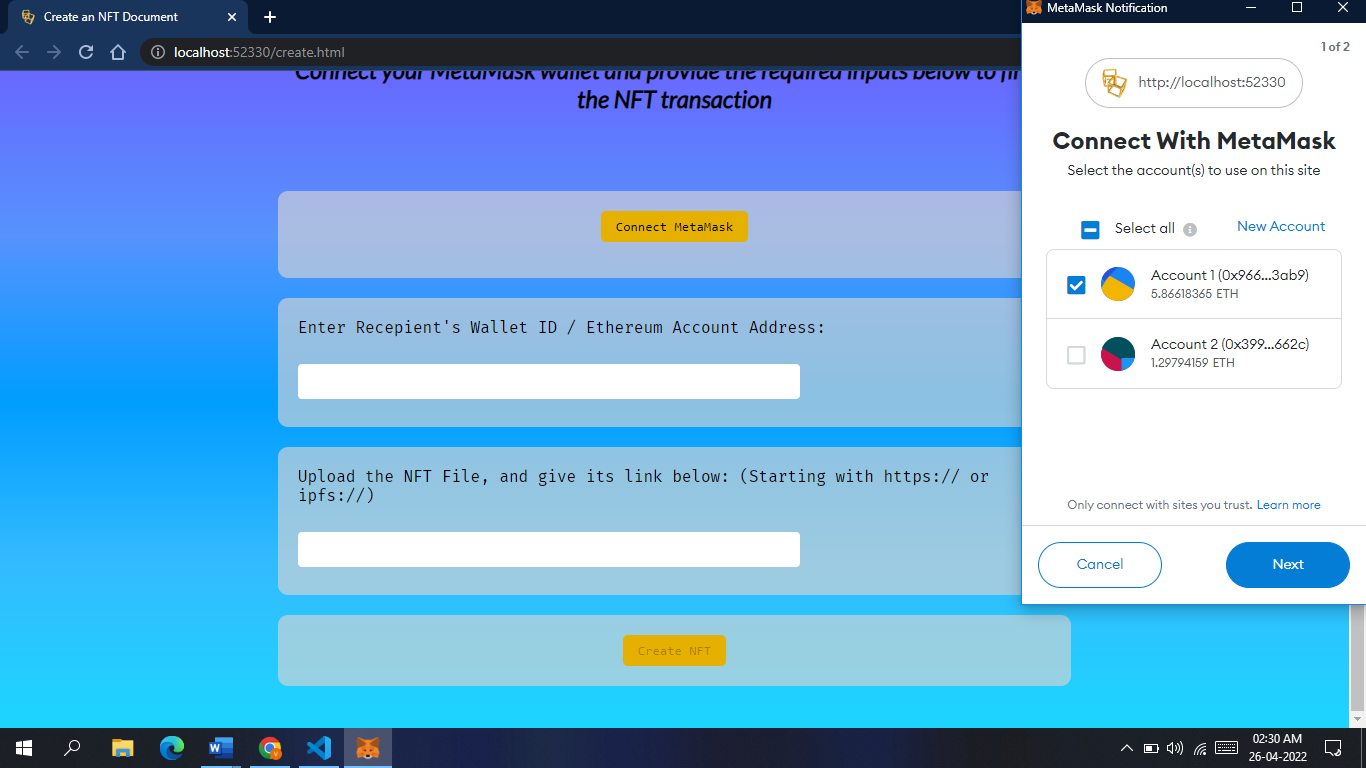


Figure 4: Selecting one account in MetaMask

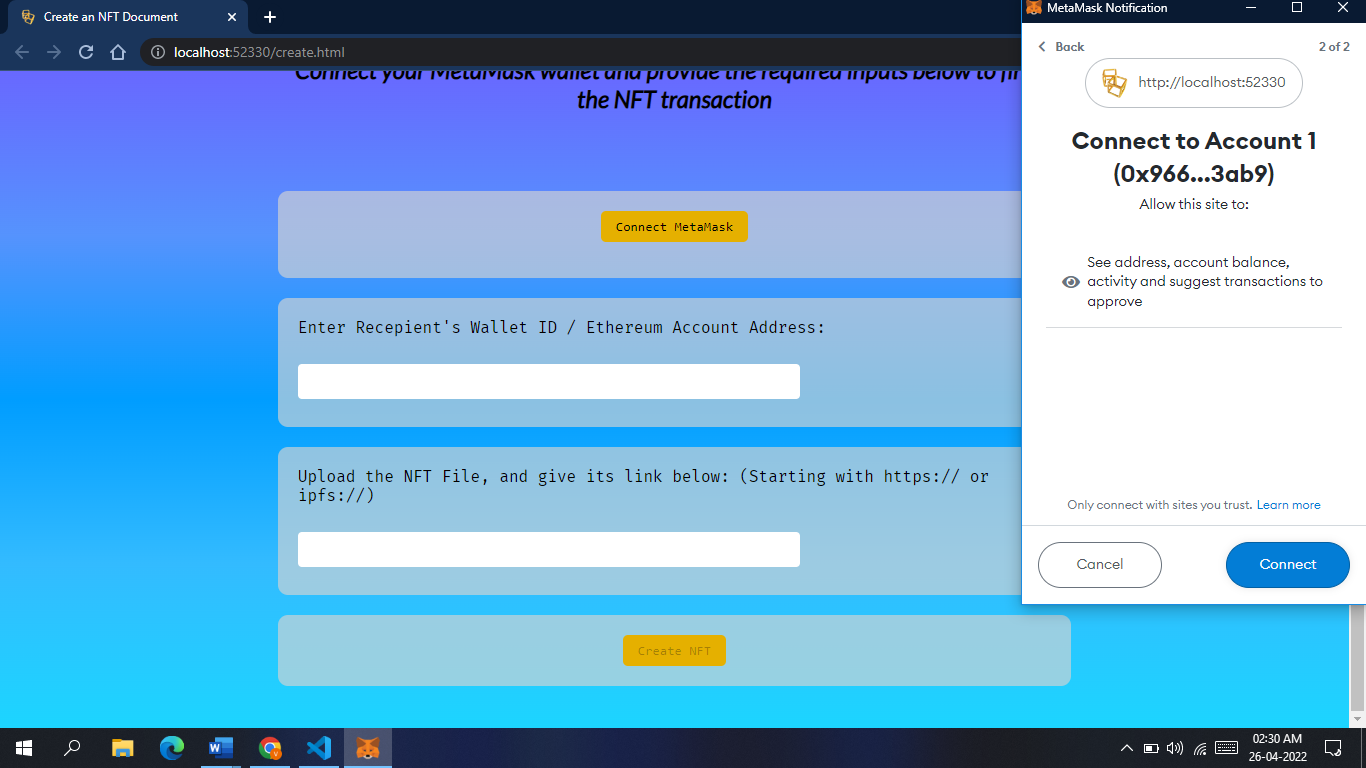


Figure 5: Giving the app permission to make transactions from Account(wallet)

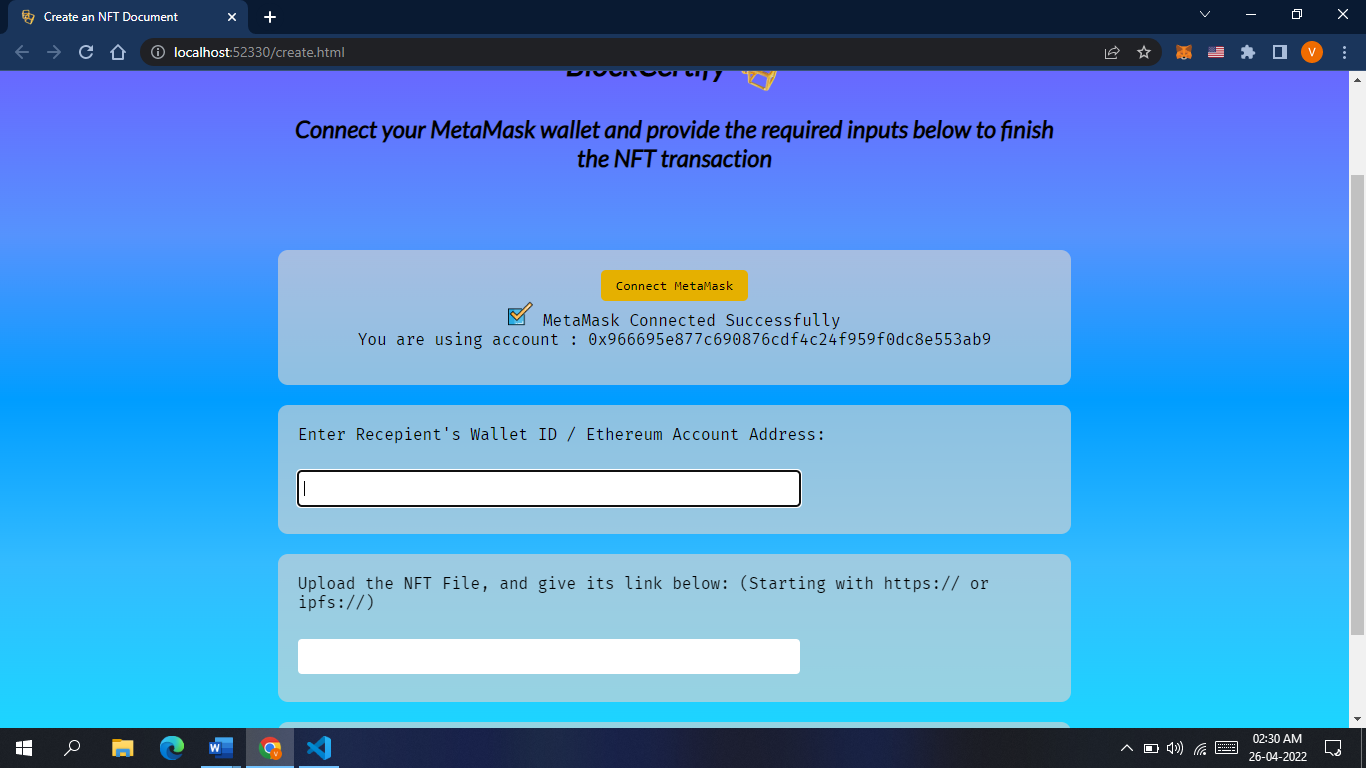


Figure 6: Prompt display that the wallet has been connected.

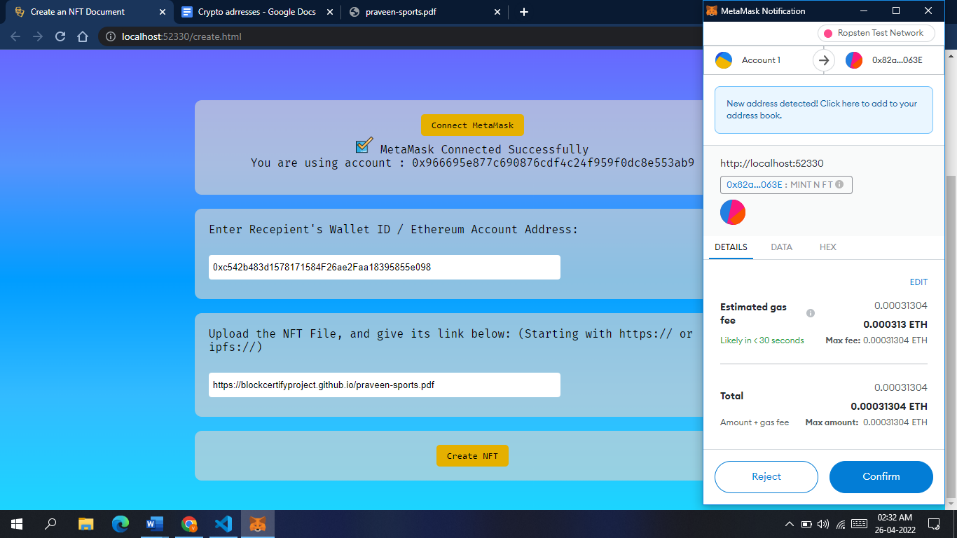
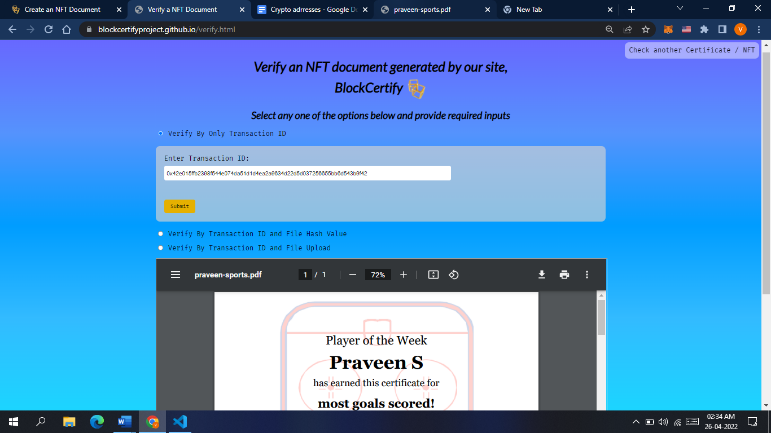


Figure 7: Upon inputting recipient wallet and file location, transaction estimated fee is shown in MetaMask



Figure 8: Upon confirming the fee, transaction is Initiated and Transaction ID is shown.

Figure 9: The transaction can be verified using Transaction ID obtained. Transaction Verification can also additionally be done with file, and file hash.

**CHAPTER 7**

**CONCLUSION**

The findings and developments in this research project indicate that the use of Blockchain Technology is effective to generate, verify and validate educational certificates in the form of NFTs (Non-Fungible Tokens). Any occurrence of a fake certificate or a fake certificate generator can be easily caught. It highly removes the communication gap between educational institutes in different geographical locations, and removes the need of third-party certificate verifying / credential evaluation / educational credential assessment companies like WES, IQAS and ICAS. This setup depends on the use of a Blockchain network, and the transaction fee incurred totally depends on the Blockchain network and its demand at the time of transaction. The transaction fee can also be reduced by reducing the transaction’s urgency of execution, by adding a delay to the total transaction time. The transaction times for the certificate generation and verification in this method have proven to be very less than the traditional methods of hard-copy documents being supplied and verified physically.

**FUTURE ENHANCEMENTS AND DISCUSSIONS**

There are a wide range of applications to NFT based certificates, and documents. Movie Tickets, Warranty Cards, Land Documents, Ownership Documents, Vehicle Registration Documents, Educational Certificates, and Jewellery and Diamond Authenticity documents, Collectibles Certificates, Government Identity Certificates etc. can all be generated as NFT’s and verified with the existing infrastructure on BlockCertify Application with minimum to no modification.

The application can be upgraded to interact with various kinds of wallets and perform transactions on multiple blockchain networks, not just Ethereum as it does now. This change can steer the application to be a one-stop location for document generation and verification at the ease of an internet-connected device.

**REFERENCES**

1. GEETHA S K et al., "Educational Certificate Verification Using Blockchain Based Framework" in Turkish Journal of Physiotherapy and Rehabilitation, May 2020
2. Shanmuga Priya R et al., “Online Certificate Validation Using Blockchain” in Special Issue Published in Int. Jnl. Of Advanced Networking & Applications (IJANA), 2019
3. Ravi Singh Lamkoti et al., “Certificate Verification using Blockchain and Generation of Transcript” in International Journal of Engineering Research & Technology (IJERT), March 2021
4. Maharshi Shah et al., “Tamper Proof Birth Certificate Using Blockchain Technology” in International Journal of Recent Technology and Engineering (IJRTE), February 2019
5. A.Gayathiri et al., “Certificate validation using blockchain” in IEEE 7th International Conference on Smart Structures and Systems ICSSS 2020, July 2020
6. Rosiyati MH Thamrin et al., “Blockchain-based Land Certificate Management in Indonesia” in ADI Journal on Recent Innovation (AJRI), February 2021
7. Neethu Gopal et al., “Survey on Blockchain Based Digital Certificate System” in International Research Journal of Engineering and Technology (IRJET), Nov 2018
8. Marco Baldi et al., “Certificate Validation through Public Ledgers and Blockchains” in First Italian Conference on Cybersecurity (ITASEC17), 2017
9. Ze Wang et al., “Blockchain-Based Certificate Transparency and Revocation Transparency” in IEEE Transactions on Dependable and Secure Computing
10. Nitin Kumavat et al., “Certificate Verification System using Blockchain” in International Journal for Research in Applied Science and Engineering Technology (IJRASET), April 2019
11. Trong Thua Huynhet al., “Issuing and Verifying Digital Certificates with Blockchain” in 2018 International Conference on Advanced Technologies for Communications (ATC) – IEEE, October 2018
12. Jiin-Chiou Cheng et al., “Blockchain and smart contract for digital certificate” in 2018 IEEE International Conference on Applied System Invention (ICASI), April 2018
13. G. Zyskind et al., “Decentralizing Privacy: Using Blockchain to Protect Personal Data” in 2015 IEEE Security and Privacy Workshops, May 2015
14. Haïdar A. M. Deenmahomed et al., “The future of university education: Examination, transcript, and certificate system using blockchain” in Computer Applications in Engineering Education, 2021
15. Neethu Gopal et al., “Smart Contract for Digital Certificate using Blockchain Technology” in International Journal for Research in Applied Science and Engineering Technology (IJRASET), May 2019
16. P.Chinnasamy, P.Deepalakshmi, V. Praveena, K.Rajakumari, P.Hamsagayathri, (2019) “Blockchain Technology: A Step Towards Sustainable Development” International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-9 Issue-2S2
17. Chinnasamy P., Vinothini C., Arun Kumar S., Allwyn Sundarraj A., Annlin Jeba S.V., Praveena V. (2021) Blockchain Technology in Smart-Cities. In: Panda S.K., Jena A.K., Swain S.K., Satapathy S.C. (eds) Blockchain Technology: Applications and Challenges. Intelligent Systems Reference Library, vol 203. Springer, Cham. https://doi.org/10.1007/978-3-030-69395-4\_11
18. A. Jain, A. Kumar Tripathi, N. Chandra and P. Chinnasamy, "Smart Contract enabled Online Examination System Based in Blockchain Network," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-7, doi: https://doi.org/10.1109/ICCCI50826.2021.9402420.
19. Ethereum.org, Non-Fungible Tokens (NFT), April 15, 2022. Accessed on April 25, 2022 [Online]. Available: https://ethereum.org/en/nft/
20. Jdourlens, Joshua Douglas, Marc-Antoine Thevenet, et al., SET UP WEB3.JS TO USE THE ETHEREUM BLOCKCHAIN IN JAVASCRIPT, April 11, 2022. Accessed on April 25, 2022 [Online]. Available: https://ethereum.org/en/developers/tutorials/set-up-web3js-to-use-ethereum-in-javascript/
21. Mark Stacey, Erik Marks, Austin Akers, et al., Provider Mitigation Guide, MetaMask, January 18, 2022. Accessed on April 24, 2022 [Online]. Available: https://docs.metamask.io/guide/provider-migration.html#replacing-window-web3
22. Henrique Centieiro, Mint an NFT and ERC-721 Smart Contract — Easy Step-by-step!, Medium Inc., Aug 18, 2021. Accessed on March 29, 2022 [Online]. Available: https://medium.com/geekculture/mint-an-nft-and-erc-721-smart-contract-easy-step-by-step-4fafff151fbe
23. Ethers.io, Provider API Keys, Ethers.io, March 27, 2022. Accessed on April 24, 2022 [Online]. Available: https://docs.ethers.io/v5/api-keys/#api-keys--getDefaultProvider