SKILL VERIFICATION CHECK SYSTEM USING ETHEREUM

CONTENTS

ABSTRACT

In the current competitive job market, employers often face challenges in verifying the accuracy of candidates' resumes and qualifications. Many candidates exaggerate or misrepresent their skills, leading to inefficiencies in the hiring process and mistrust among employers. Traditional methods of skill verification are time-consuming, prone to bias, and lack transparency, making it difficult for employers to assess the actual abilities of potential employees. This results in poor hiring decisions, wasted resources, and a lack of accountability in candidates' skill claims.

To address these challenges, the Decentralized Skill Verification System introduces a trustworthy, transparent, and efficient solution for skill verification by leveraging blockchain technology.

Key Objectives of the System:

- Trustworthy Skill Verification: The system will allow individuals to verify their skills through endorsements from peers, industry professionals, and recognized institutions. By securing this data on a blockchain, the system ensures the credibility of skill claims, making it nearly impossible for candidates to misrepresent their qualifications.
- Efficient Hiring Process: Employers will gain access to verified skill profiles, allowing them to make more informed and objective hiring decisions based on factual data. This eliminates the guesswork and subjectivity that often accompany traditional resume screenings.
- Peer and Professional Endorsements: Individuals can substantiate their skills through endorsements. These endorsements, recorded on the blockchain, add an additional layer of trust, as they are backed by professionals or institutions with proven expertise in the relevant fields.
- Decentralization and Transparency: By utilizing blockchain's immutable ledger, the system creates a transparent, decentralized process for skill validation. This means that once a skill has been verified and recorded, it cannot be altered or tampered with, ensuring the integrity of the data.
- Accountability and Honesty: The platform promotes a culture of accountability by encouraging candidates to be honest about their skills. A transparent endorsement process pushes individuals to uphold their credibility, fostering trust among employers and creating a more reliable and fair job market.

By integrating blockchain technology, this decentralized system addresses the common challenges of traditional skill verification processes, providing a more efficient, transparent, and reliable method for both job seekers and employers. It encourages integrity in skill claims, improves the hiring process, and builds a trustworthy job market for all parties involved.

CHAPTER 1

INTRODUCTION

1.1 Skill Verification System

In today's competitive landscape, verifying and showcasing skills is crucial for both professionals and employers. The Skill Verification System is a blockchain-based platform designed to ensure the authenticity and validity of skills and qualifications. By leveraging blockchain's decentralized and immutable properties, this project aims to provide a secure, transparent, and tamper-proof mechanism for verifying user-submitted skills.

This system allows users to register, submit proof of skills through documents, and have their credentials verified by an administrator. Once verified, the skills become part of an immutable record, ensuring that they are trusted and verifiable by any third party. The use of blockchain guarantees that all skill records are transparent and cannot be altered after verification, fostering trust in professional credentials.

1.2 Motivation

The primary motivation behind the Skill Verification System is the increasing concern over fraudulent claims and unverified credentials in job applications. Many individuals face challenges in proving the authenticity of their skills, while employers struggle to trust unverified claims. This project aims to create a reliable platform where users can securely submit their skill documents for verification, and organizations can trust that the verified information is accurate.

By utilizing blockchain technology, the system eliminates the need for a centralized authority, thereby reducing the risks of tampering and providing users full control over their credentials. This transparency will empower professionals by giving them an immutable way to prove their skills, while simplifying the hiring process for employers.

1.3 Problem Statement

The current process of verifying professional skills is inefficient, lacks transparency, and is susceptible to fraud. With many individuals falsifying credentials and employers finding it difficult to validate claims, the need for a secure, reliable solution is evident. Additionally, centralized systems that manage skill verification are prone to errors, manipulation, and data breaches.

To solve this, the Skill Verification System introduces a decentralized platform that records, verifies, and validates skills on the blockchain. Users can submit documents proving their skills, which will be verified by administrators. Once verified, these skills will be added to the blockchain, creating a tamper-proof and easily accessible record of the user's qualifications.

1.4 Objective

The objectives of the system are:

1. **Build a Secure Verification System:** Create a blockchain-based platform that ensures the security and authenticity of verified skills.

2. **Decentralized Data Management:** Store verified skills in a decentralized manner to ensure tamper-proof and transparent records.

3. **User-Friendly Platform:** Develop an intuitive interface where users can register, upload documents, and manage their verified skills with ease.

4. **Empower Users:** Provide users with full control over their skill records, enabling them to share their credentials with potential employers or other parties.

5. **Admin Verification:** Implement an admin-controlled verification process to ensure that submitted skills are thoroughly checked before being recorded on the blockchain.

1.5 Scope

The scope of the system includes the development of a decentralized platform that will allow users to securely upload documents proving their skills. Administrators will be responsible for verifying these documents, after which the verified credentials will be stored on the blockchain. This will ensure that the records cannot be tampered with and are easily verifiable by third parties.

The platform will also feature a user-friendly interface where individuals can manage their skill profiles and submit new qualifications for verification. Additionally, the system will provide APIs for third parties, such as potential employers, to verify the credentials on the blockchain without needing to access sensitive information.

The system will ultimately create a more secure, efficient, and reliable way to verify skills, benefiting both users and organizations.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 Functional Requirements

- **User Registration and Login**

  Description: Users should be able to register by providing basic details such as name, email, and password. After registration, they can log in to access the system.

  Actors: Users (Job Seekers, Employees) and Admins.

  Precondition: Users should have valid credentials to log in.

  Postcondition: Successfully authenticated users can access their dashboard, and admins can view user submissions.

- **Skill Submission**

  Description: Users should be able to submit skills with supporting documents (e.g., certificates, proof of experience). Each submission will be stored on the platform until verified.

  Actors: Users (Job Seekers).

  Precondition: Users are logged in.

  Postcondition: Users can view their submitted skills, pending for verification.

- **Admin Skill Verification**

  Description: Admins should be able to view submitted skills, verify the attached documents, and either accept or reject the verification.

  Actors: Admins.

  Precondition: Admins have access to the admin panel.

  Postcondition: Once verified, the skill is recorded on the blockchain, or the user is notified of the rejection with reasons.

- **Blockchain Record Storage**

  Description: Upon successful verification, the user's skill is stored immutably on the blockchain. This ensures that the verified skill is tamper-proof and can be viewed publicly.

Actors: Blockchain system (Ethereum).

Precondition: A smart contract for skill verification is deployed on the blockchain.

Postcondition: Verified skills are recorded as transactions on the blockchain.

- **View Verified Skills**

  Description: Both users and third-party verifiers (e.g., employers) should be able to view a list of verified skills linked to the user's profile. These records will be accessible via the blockchain.

  Actors: Users, Third-party verifiers.

  Precondition: Skills must be verified and stored on the blockchain.

  Postcondition: Skills can be viewed via a public or restricted view.

- Document Storage on IPFS

  Description: All skill-related documents (e.g., certificates) will be stored on a decentralized file storage system like IPFS to ensure secure and scalable document handling.

  Actors: Users, Admins.

  Precondition: IPFS nodes must be set up for document storage.

  Postcondition: Documents will be securely stored on IPFS with a reference (hash) stored on the blockchain.

2.1 Non-Functional Requirements

- Blockchain Security: All verified skills should be stored on the blockchain to ensure immutability and prevent tampering.
- Authentication: Secure login mechanisms should be implemented to prevent unauthorized access to user and admin accounts.
- Document Security: All user-uploaded documents should be encrypted and stored on IPFS to ensure data integrity and security.
- Blockchain Transactions: Transactions related to storing verified skills on the blockchain must be optimized for performance to reduce latency.
- Scalability: The system must scale to handle a growing number of users, submissions, and verifications.
- Response Time: The system should provide a responsive user experience with minimal lag for user interactions and blockchain transactions.

The user interface must be intuitive and user-friendly, making it easy for users to register, log in, submit, and manage their skills.

Admins should have a clean and organized dashboard to manage and verify submissions efficiently.

- Reliability- The system must be highly reliable, with minimal downtime, to ensure users can continuously submit and verify skills. All verified data stored on the blockchain must be accessible 24/7 without risk of data loss.
- Privacy- Personal user information, including credentials and documents, must be handled securely and should not be exposed to unauthorized parties. Only verified parties should have access to user-submitted skills and documents.

2.3 Hardware Requirements

- Minimum: 8 GB RAM, 4 vCPUs, 200 GB SSD (for hosting the backend and blockchain nodes).
- Recommended: 16 GB RAM, 8 vCPUs, 500 GB SSD.
- User Machines: For accessing the platform, any device with modern web browsers like Google Chrome, Firefox, or Edge is sufficient.
- A stable internet connection is required for both users and admins.

2.4 Software Requirements

- Operating System: Windows, Linux, or macOS for hosting the backend and blockchain nodes.
- Blockchain Environment: Ganache for local blockchain development and testing.
- Smart Contracts: Solidity (for writing smart contracts).
- Frontend: HTML, CSS, JavaScript (for creating the user and admin interfaces).
- Decentralized Storage: IPFS (for document storage).
- Version Control: Git (for code versioning and collaboration).

2.5 Technology Stack

- Frontend- HTML/CSS/JavaScript: Basic structure and styling of the user interface.
- Web3.js: For interacting with the Ethereum blockchain.
- IPFS(Interplanetary File System) Client: For handling file uploads and retrievals on IPFS.
- Backend- Smart Contracts (Solidity): For managing the verification process and recording

verified skills on the Ethereum blockchain.

- Blockchain

  Ethereum: Main blockchain network for storing verified skill records immutably.

  Ganache: Local Ethereum blockchain for testing and development.

  Truffle: Framework for compiling, testing, and deploying smart contracts.

- Storage- IPFS: Decentralized file system for storing user-submitted documents.

- Integration Requirements- Smart Contract Deployment: A smart contract needs to be deployed on an Ethereum-compatible blockchain. For development and testing, Ganache will be used, while deployment on a public blockchain will be the end goal.

  IPFS Integration: The platform needs to integrate with IPFS to store documents securely and retrieve them using unique content identifiers (hashes).

  Web3 Integration: JavaScript frontend must integrate with Web3.js to interact with the smart contract and manage blockchain transactions.

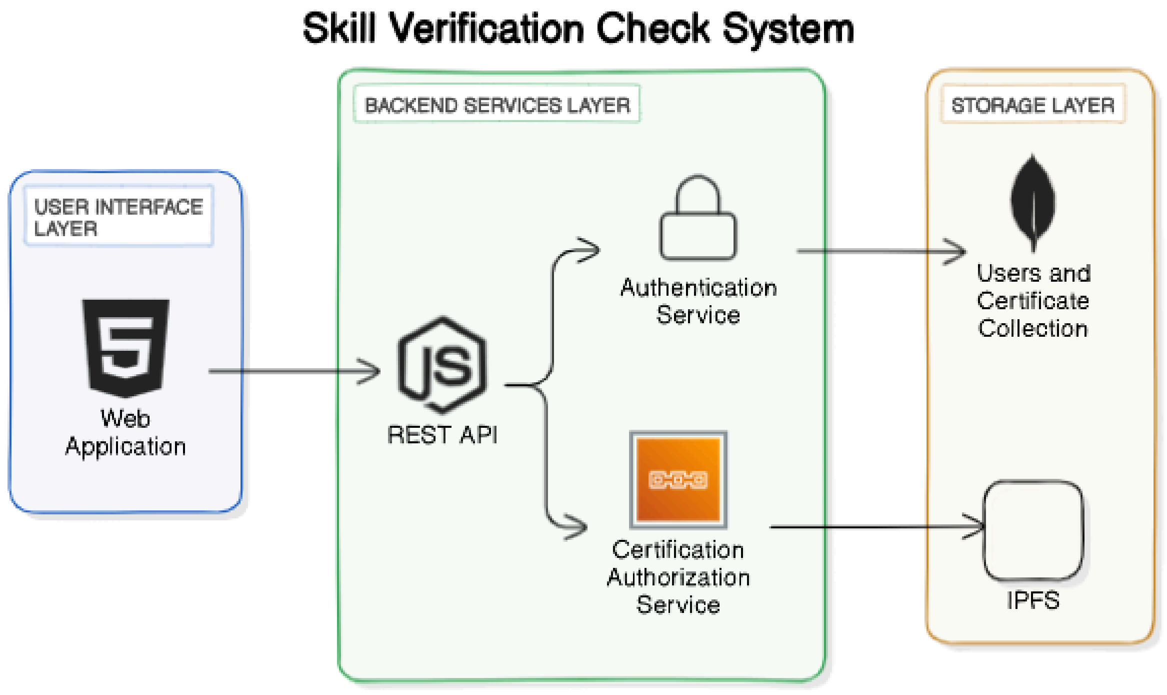
CHAPTER 3

SYSTEM ARCHITECTURE

3.1 System Architecture

The overall architecture of the Skill Verification System can be divided into three main layers:

- User Interface Layer
- Backend Services Layer
- Storage Layer



User Interface Layer

The User Interface (UI) layer is responsible for user interactions with the system. It consists of:

- **Web Application**: Built using frameworks like React or Angular, this application allows users to perform the following actions:
  - Register and log in to their accounts.
  - Upload skill certifications.
  - View and verify other users' certifications.

Backend Services Layer

The Backend Services layer handles business logic, API requests, and data processing. It is composed of:

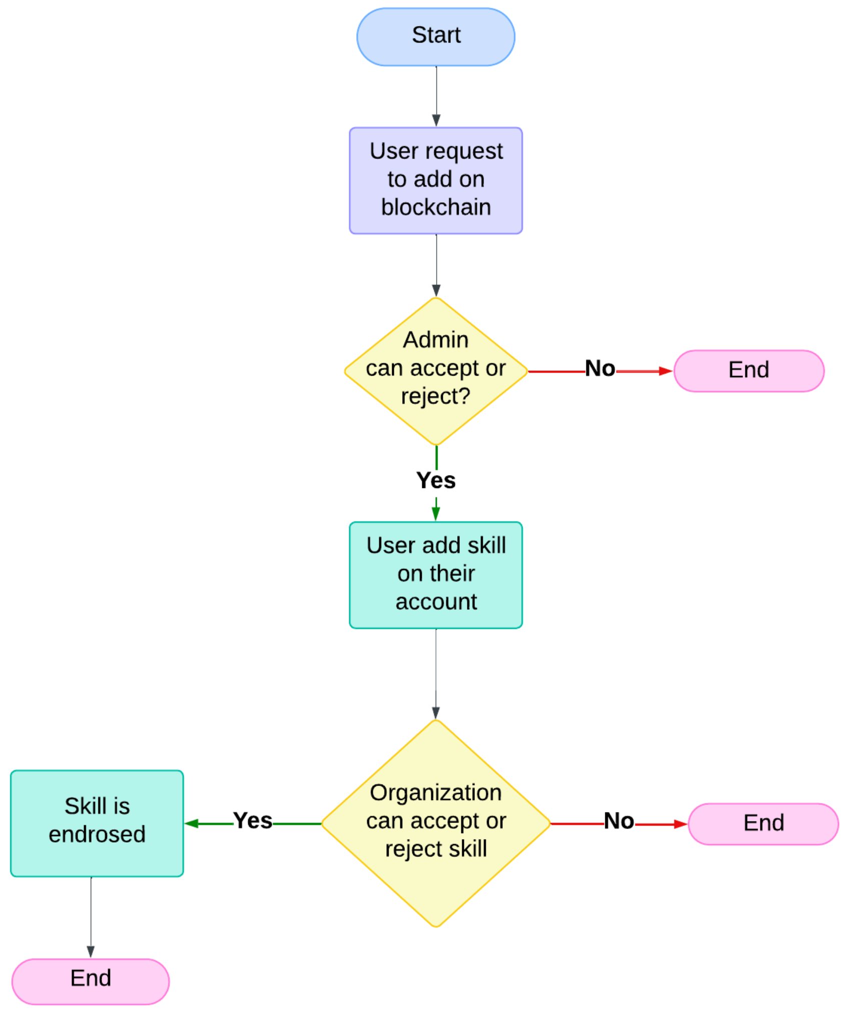
- **REST API (Node.js with Express)**: The backend server processes requests from the frontend, manages authentication, and interacts with the database and IPFS. It exposes various endpoints for:
    - User registration and authentication.
    - Uploading and retrieving certifications.
- **Authentication Service**: This service manages user authentication using JSON Web Tokens (JWT) to ensure secure communication between the frontend and backend.
- **Certification Management Service**: Responsible for handling certification uploads, which involves:
    - Interfacing with IPFS to store certification documents.
    - Storing metadata in the database for easy retrieval.

Storage Layer

The Storage layer is divided into two components:

-
    - **Users Collection**: Stores user information such as usernames and hashed passwords.
    - **Certifications Collection**: Contains information about each certification, including user IDs, certification details, and IPFS hashes.
- **IPFS**: A decentralized storage solution where certification documents are uploaded. The IPFS hash is saved in the database, allowing for secure and tamper-proof storage of certification files.

3.2 System Flow Diagram


Start
User request to add on blockchain
Admin can accept or reject?
No
End
Yes
User add skill on their account
Organization can accept or reject skill
Yes
Skill is endrosed
No
End
End

The data flow in the Skill Verification System is as follows:

1. **User Registration and Login**: Users register and log in through the web application, which communicates with the REST API for authentication.
2. **Uploading Certifications**: Upon successful authentication, users can upload their skill certifications. The web application sends the file to the backend, which:
    - Uploads the file to IPFS.
    - Receives the IPFS hash and saves it in the database along with certification metadata.
3. **Certification Verification**: Users can view their certifications or verify others by querying the database. The frontend retrieves certification details from the backend, including the

IPFS link for viewing the uploaded document.

3.4 External Services Integration

The Skill Verification System can integrate with external services to enhance functionality:

- **IPFS API**: Interacts with IPFS for storing and retrieving certification documents. Or
- **Email Service (e.g., SendGrid)**: Used for sending confirmation emails during registration and notifications related to certification updates.

CHAPTER 4

TOOLS AND TECHNOLOGIES

4.1 Frontend Technologies

HTML (HyperText Markup Language)

- **Description**: The standard markup language for creating web pages.
- **Role**: HTML structures the web application's interface, allowing for user input through forms for certification uploads and user registrations.

CSS (Cascading Style Sheets)

- **Description**: A style sheet language used for describing the presentation of a document written in HTML.
- **Role**: CSS is employed to style the application, ensuring a responsive design that enhances user experience across devices.

JavaScript

- **Description**: A programming language that enables interactive web pages.
- **Role**: JavaScript handles client-side scripting, allowing users to interact with the application seamlessly, manage form submissions, and dynamically display content.

Web3.js

- **Description**: A JavaScript library that allows interaction with the Ethereum blockchain.
- **Role**: Web3.js is used for managing smart contracts and handling blockchain-related interactions, enabling features like wallet connection and transaction management.

IPFS HTTP Client

- **Description**: A JavaScript client for interfacing with IPFS (InterPlanetary File System).
- **Role**: This library is essential for uploading files (certifications) to IPFS and retrieving IPFS hashes, which are used for accessing stored documents.

4.2 Backend Technologies

Node.js

- **Description**: A JavaScript runtime built on Chrome's V8 engine, allowing server-side execution of JavaScript.

- **Role**: Node.js is utilized to build the backend server, managing API requests, user sessions, and integration with decentralized storage solutions.

Express.js

- **Description**: A minimal web application framework for Node.js that simplifies API development.
- **Role**: Express.js is used to create RESTful APIs that facilitate user authentication, certification management, and interaction with the IPFS service.

4.3 Decentralized Storage Technologies

IPFS (Inter Planetary File System)

- **Description**: A peer-to-peer distributed file system that allows the storage and sharing of files in a decentralized manner.
- **Role**: IPFS is employed to securely store certification documents, ensuring high availability and integrity. Users can retrieve documents using their unique IPFS hashes.

4.4 Development Tools

Visual Studio Code

- **Description**: A lightweight yet powerful source code editor with built-in support for JavaScript and Node.js.
- **Role**: Visual Studio Code serves as the primary integrated development environment (IDE) for writing, debugging, and testing the application's codebase.

Git

- **Description**: A version control system for tracking changes in source code during software development.
- **Role**: Git is used to manage version control for the project's codebase, facilitating collaboration among team members through platforms like GitHub.

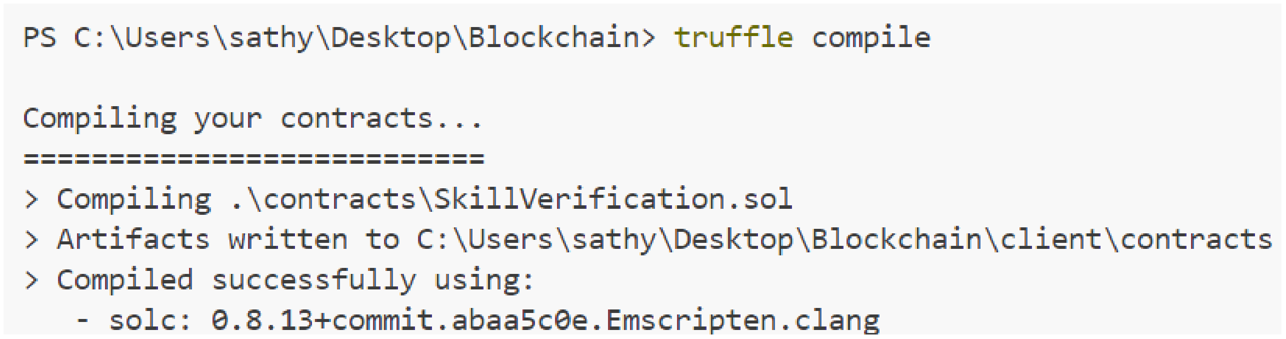
Postman

- **Description**: An API development tool that allows for testing and documenting APIs.
- **Role**: Postman is employed to test API endpoints, ensuring they function correctly and respond as expected during development.

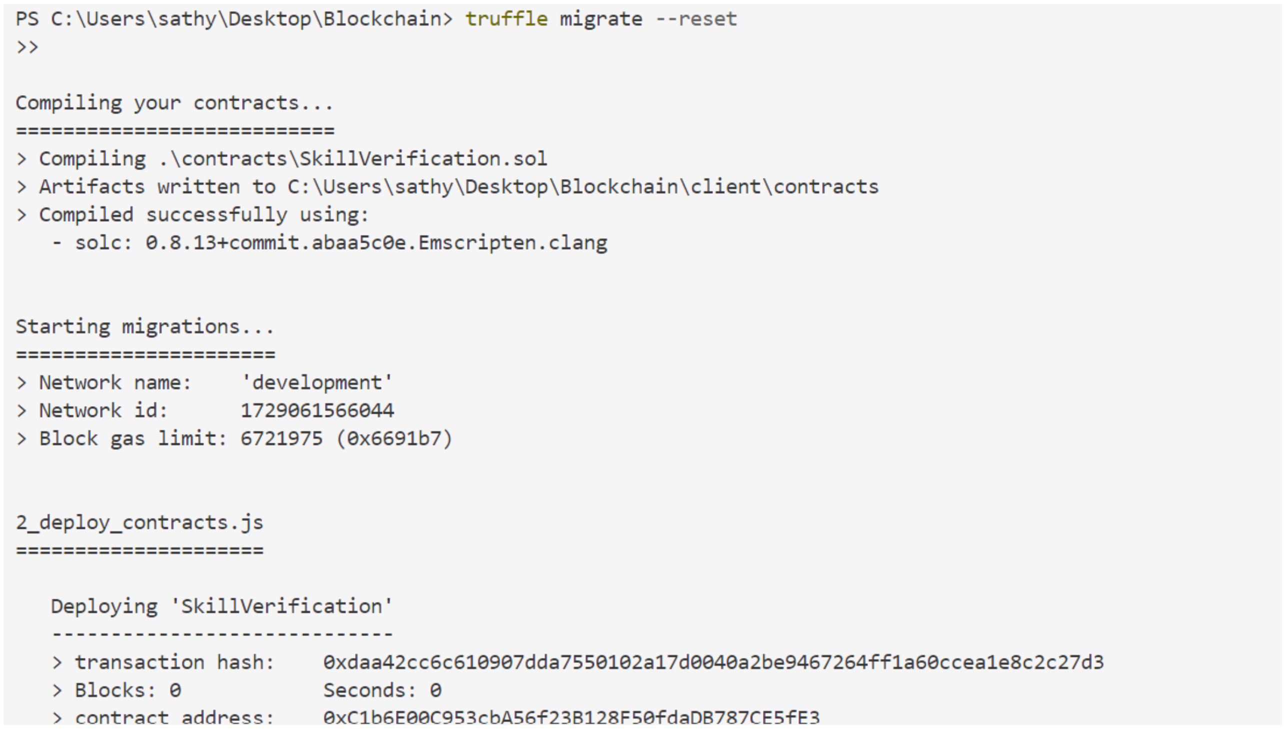
CHAPTER 5

IMPLEMENTATION

Compile smart contracts- Truffle

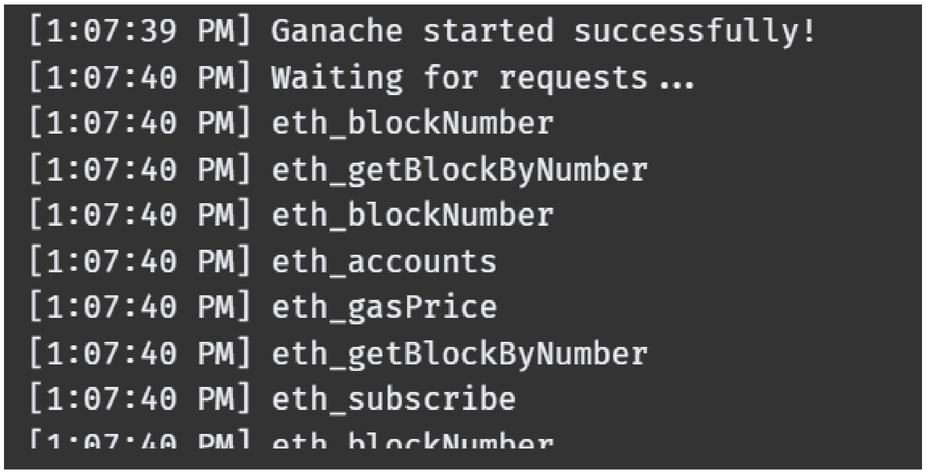


```
PS C:\Users\sathy\Desktop\Blockchain> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\SkillVerification.sol
> Artifacts written to C:\Users\sathy\Desktop\Blockchain\client\contracts
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang
```



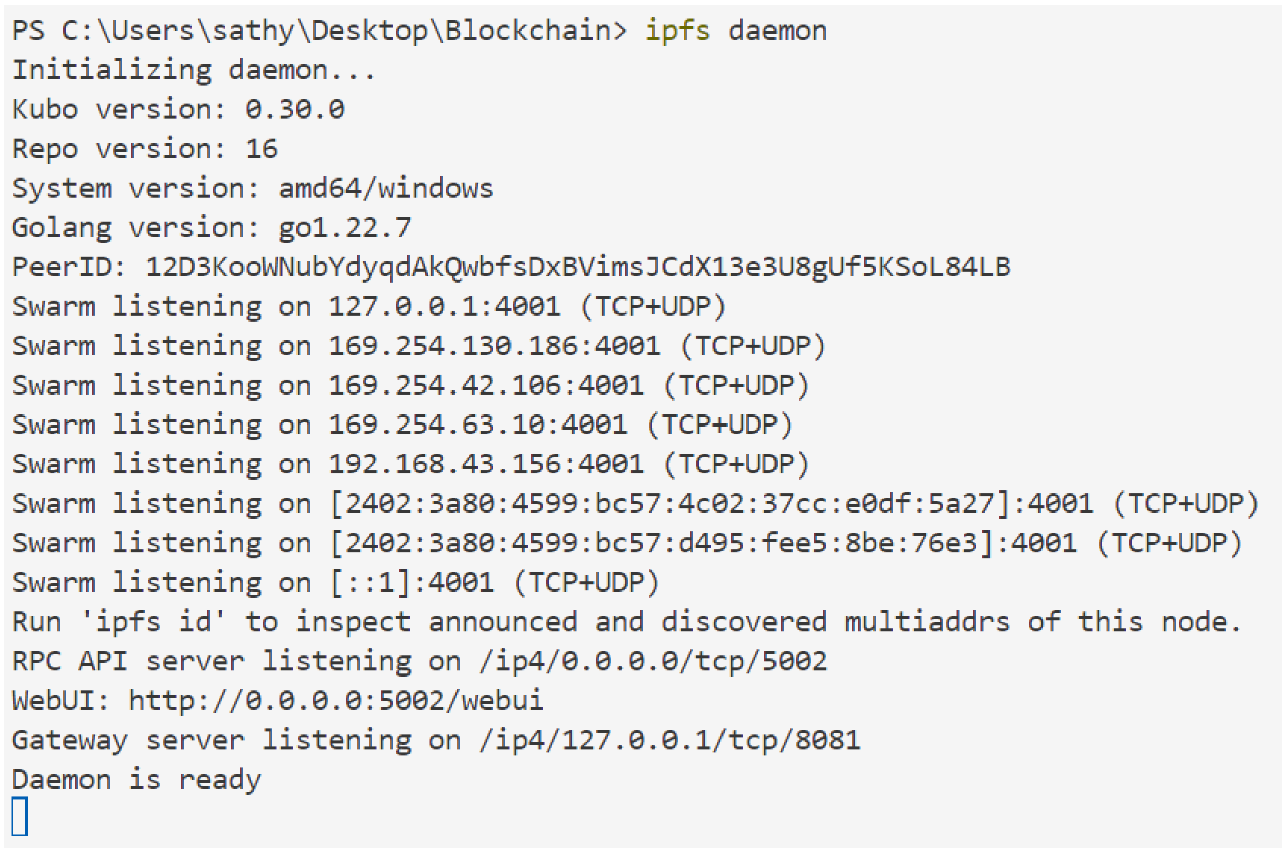
```
PS C:\Users\sathy\Desktop\Blockchain> truffle migrate --reset
>>

Compiling your contracts...
===========================
> Compiling .\contracts\SkillVerification.sol
> Artifacts written to C:\Users\sathy\Desktop\Blockchain\client\contracts
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      1729061566044
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Deploying 'SkillVerification'
   -----------------------------
   > transaction hash:    0xdaa42cc6c610907dda7550102a17d0040a2be9467264ff1a60ccea1e8c2c27d3
   > Blocks: 0            Seconds: 0
   > contract address:    0xC1b6E00C953cbA56f23B128F50fdaDB787CE5fE3
```

Deploy smart contracts to ganache and start IPFS:

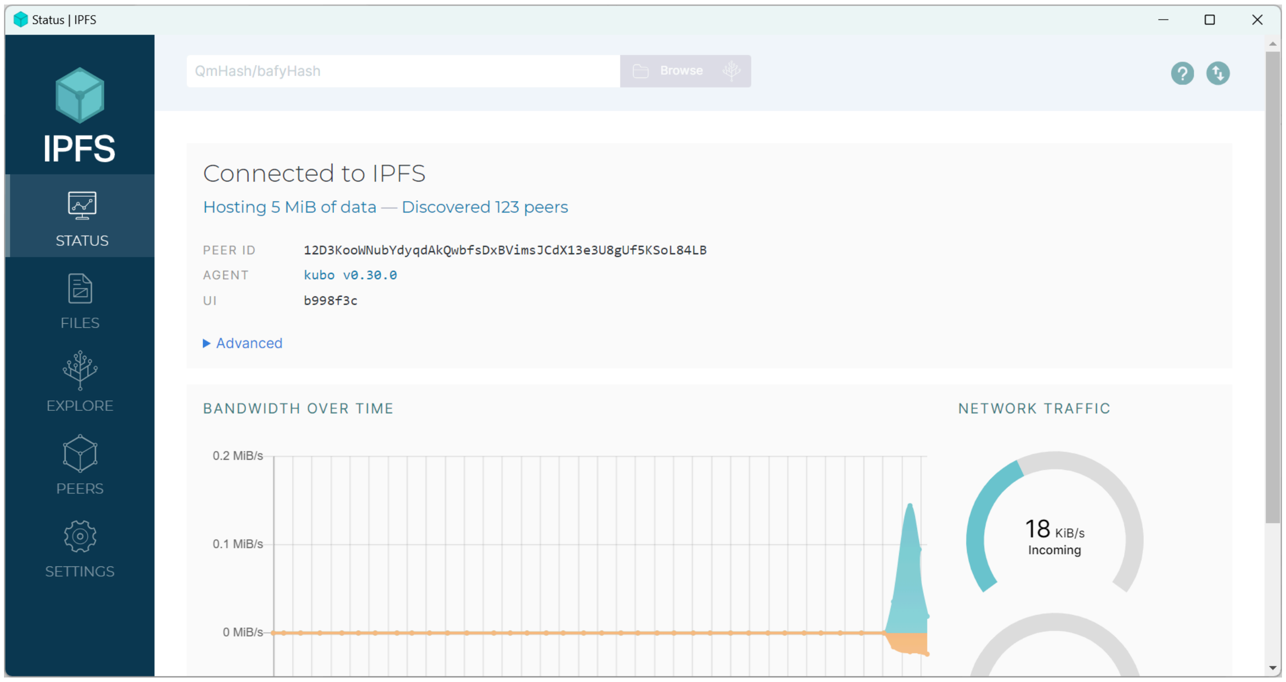


```
[1:07:39 PM] Ganache started successfully!
[1:07:40 PM] Waiting for requests...
[1:07:40 PM] eth_blockNumber
[1:07:40 PM] eth_getBlockByNumber
[1:07:40 PM] eth_blockNumber
[1:07:40 PM] eth_accounts
[1:07:40 PM] eth_gasPrice
[1:07:40 PM] eth_getBlockByNumber
[1:07:40 PM] eth_subscribe
[1:07:40 PM] eth_blockNumber
```

IPFS Daemon:



```
PS C:\Users\sathy\Desktop\Blockchain> ipfs daemon
Initializing daemon...
Kubo version: 0.30.0
Repo version: 16
System version: amd64/windows
Golang version: go1.22.7
PeerID: 12D3KooWNubYdyqdAkQwbfsDxBVimsJCdX13e3U8gUf5KSoL84LB
Swarm listening on 127.0.0.1:4001 (TCP+UDP)
Swarm listening on 169.254.130.186:4001 (TCP+UDP)
Swarm listening on 169.254.42.106:4001 (TCP+UDP)
Swarm listening on 169.254.63.10:4001 (TCP+UDP)
Swarm listening on 192.168.43.156:4001 (TCP+UDP)
Swarm listening on [2402:3a80:4599:bc57:4c02:37cc:e0df:5a27]:4001 (TCP+UDP)
Swarm listening on [2402:3a80:4599:bc57:d495:fee5:8be:76e3]:4001 (TCP+UDP)
Swarm listening on [::1]:4001 (TCP+UDP)
Run 'ipfs id' to inspect announced and discovered multiaddrs of this node.
RPC API server listening on /ip4/0.0.0.0/tcp/5002
WebUI: http://0.0.0.0:5002/webui
Gateway server listening on /ip4/127.0.0.1/tcp/8081
Daemon is ready
```



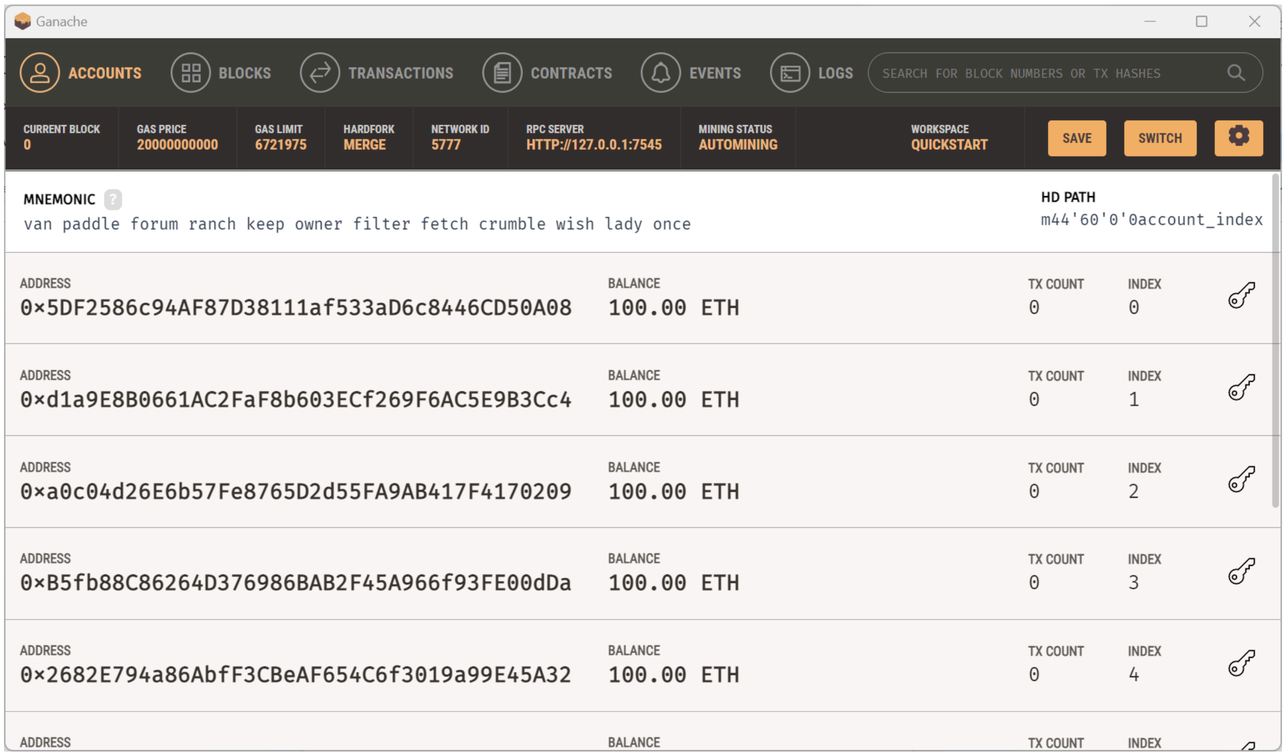
Connected to IPFS

Hosting 5 MiB of data — Discovered 123 peers

PEER ID	12D3KooWNubYdyqdAkQwbfsDxBVimsJCdX13e3U8gUf5KSoL84LB
AGENT	kubo v0.30.0
UI	b998f3c

▶ Advanced

Connect ganache with metamask:

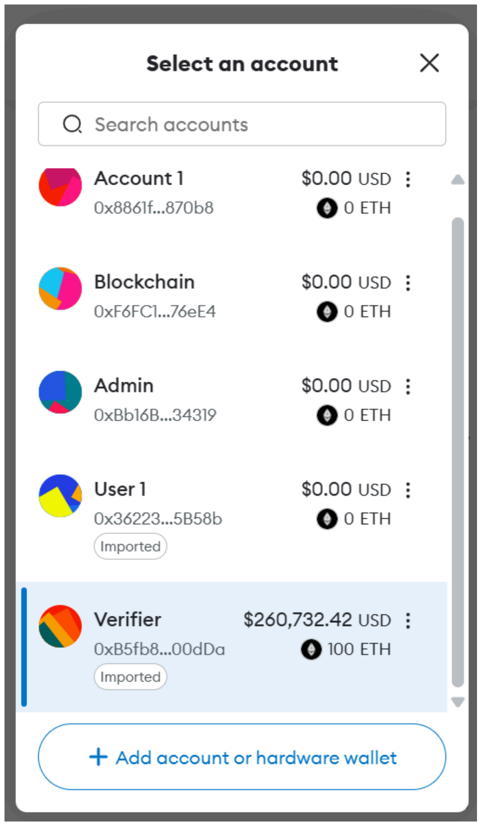


CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
0	20000000000	6721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART

MNEMONIC
van paddle forum ranch keep owner filter fetch crumble wish lady once

HD PATH
m44'60'0'0account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x5DF2586c94AF87D38111af533aD6c8446CD50A08	100.00 ETH	0	0
0xd1a9E8B0661AC2FaF8b603ECf269F6AC5E9B3Cc4	100.00 ETH	0	1
0xa0c04d26E6b57Fe8765D2d55FA9AB417F4170209	100.00 ETH	0	2
0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	100.00 ETH	0	3
0x2682E794a86AbfF3CBeAF654C6f3019a99E45A32	100.00 ETH	0	4

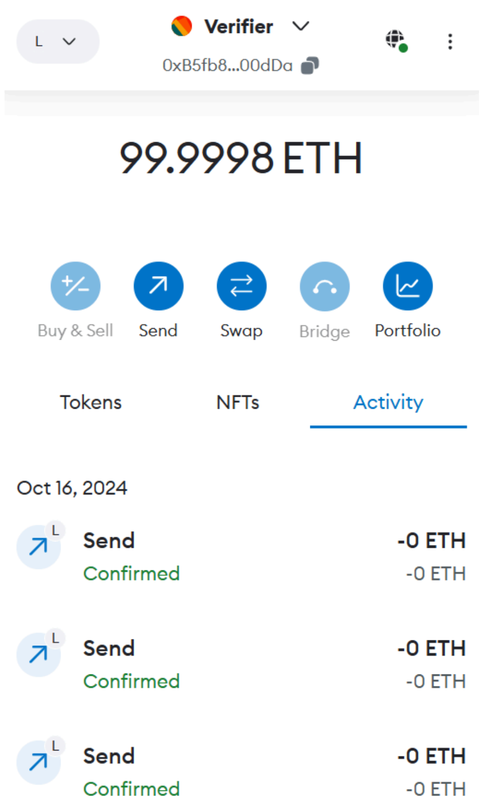


Select an account

Search accounts

Account 1	0x8861f...870b8	$0.00 USD, 0 ETH
Blockchain	0xF6FC1...76eE4	$0.00 USD, 0 ETH
Admin	0xBb16B...34319	$0.00 USD, 0 ETH
User 1	0x36223...5B58b (Imported)	$0.00 USD, 0 ETH
Verifier	0xB5fb8...00dDa (Imported)	$260,732.42 USD, 100 ETH

+ Add account or hardware wallet



Verifier
0xB5fb8...00dDa

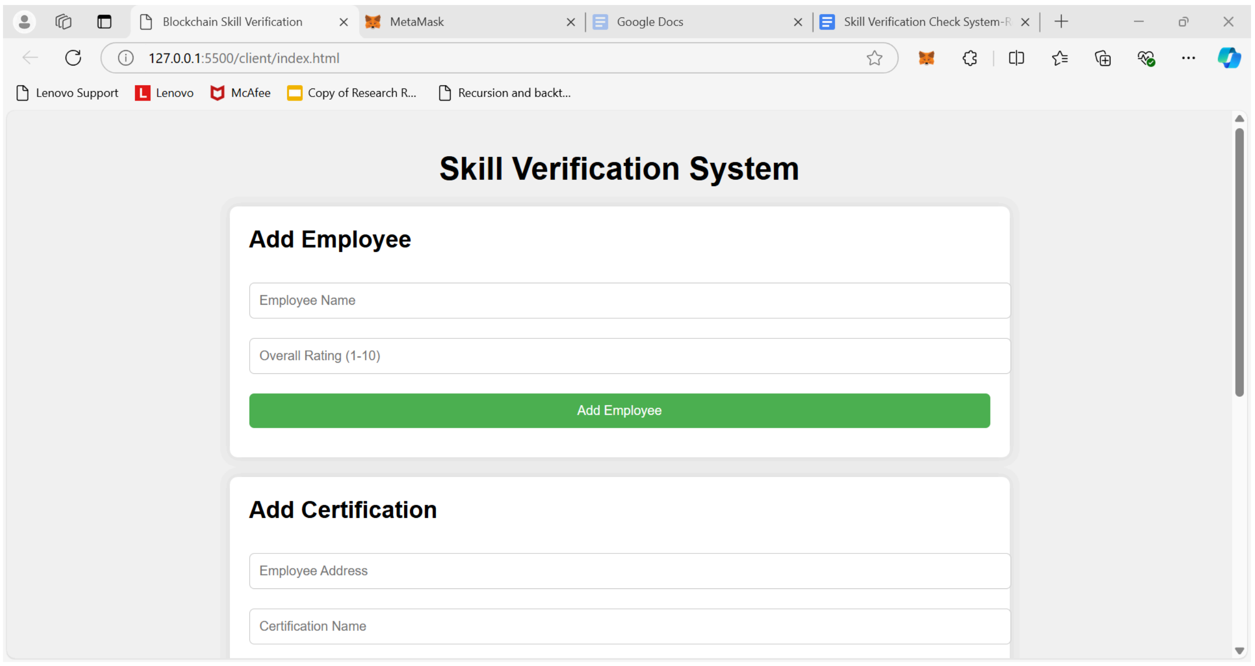
99.9998 ETH

Oct 16, 2024

Send Confirmed -0 ETH -0 ETH

Send Confirmed -0 ETH -0 ETH

Send Confirmed -0 ETH -0 ETH



Blockchain Skill Verification
MetaMask
Google Docs
Skill Verification Check System-R
127.0.0.1:5500/client/index.html
Lenovo Support
Lenovo
McAfee
Copy of Research R...
Recursion and backt...

Skill Verification System

Add Employee

Employee Name

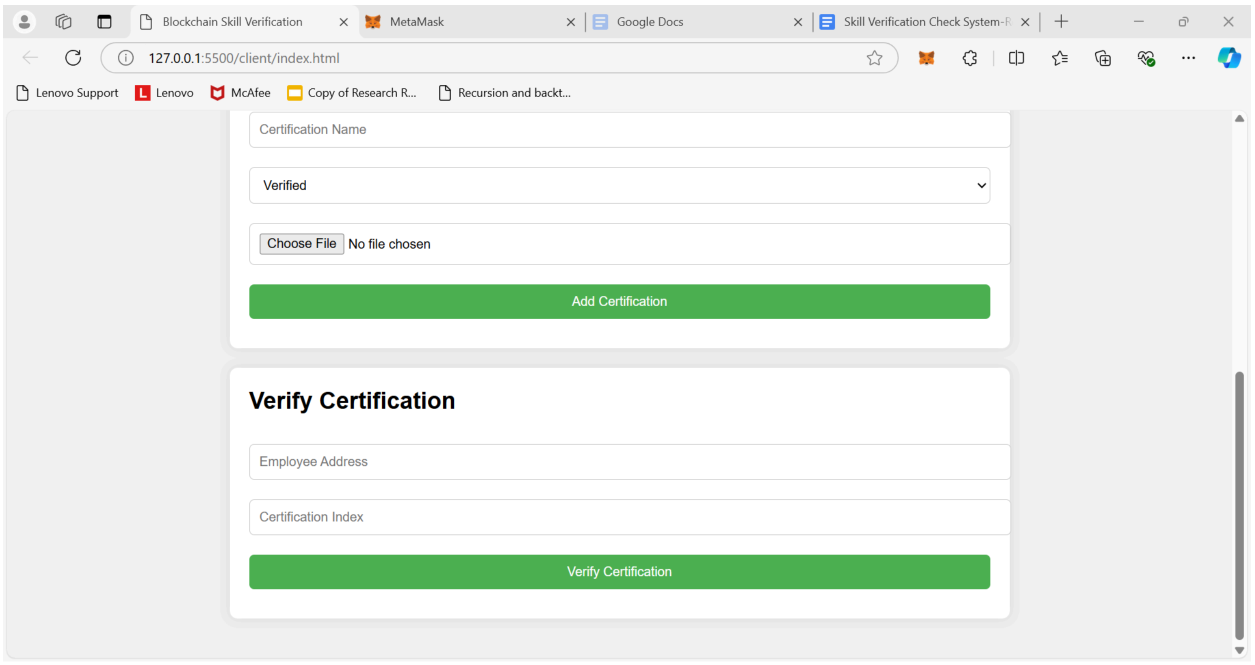
Overall Rating (1-10)

Add Employee

Add Certification

Employee Address

Certification Name



Blockchain Skill Verification

MetaMask

Google Docs

Skill Verification Check System-R

127.0.0.1:5500/client/index.html

Lenovo Support | Lenovo | McAfee | Copy of Research R... | Recursion and backt...

Certification Name

Verified

Choose File No file chosen

Add Certification

Verify Certification

Employee Address

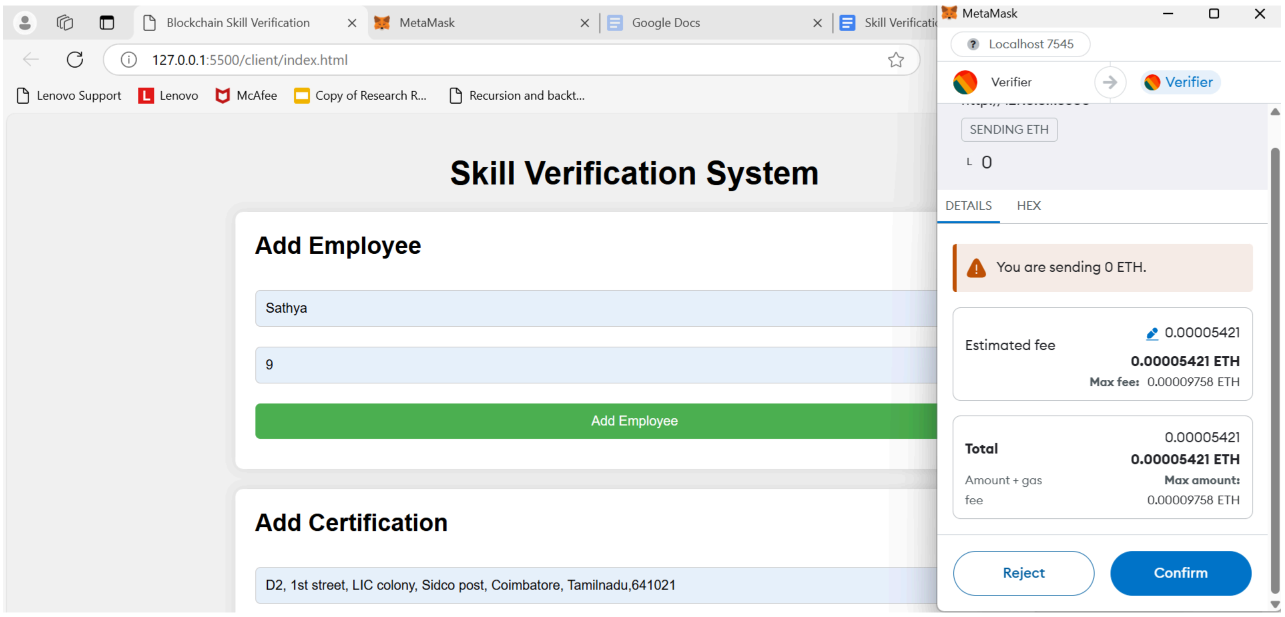
Certification Index

Verify Certification

CHAPTER 6

RESULTS AND OBSERVATION

Add Employee


Skill Verification System

Add Employee

Sathya

9

Add Employee

Add Certification

D2, 1st street, LIC colony, Sidco post, Coimbatore, Tamilnadu,641021

MetaMask

Localhost 7545

Verifier → Verifier

SENDING ETH

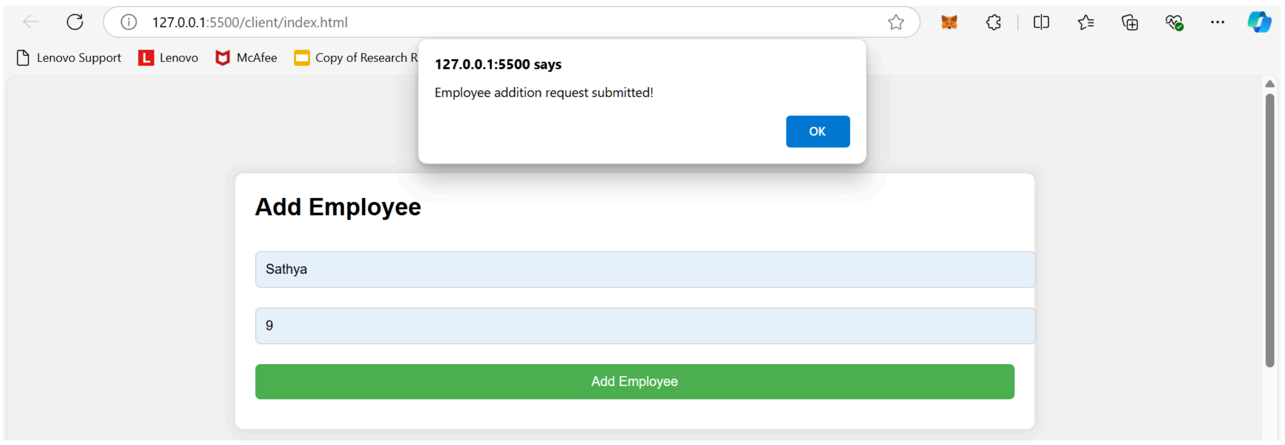
0

You are sending 0 ETH.

Estimated fee	0.00005421
	0.00005421 ETH
	Max fee: 0.00009758 ETH
Total	0.00005421
	0.00005421 ETH
Amount + gas fee	Max amount: 0.00009758 ETH

Reject Confirm

Confirm


127.0.0.1:5500 says

Employee addition request submitted!

OK

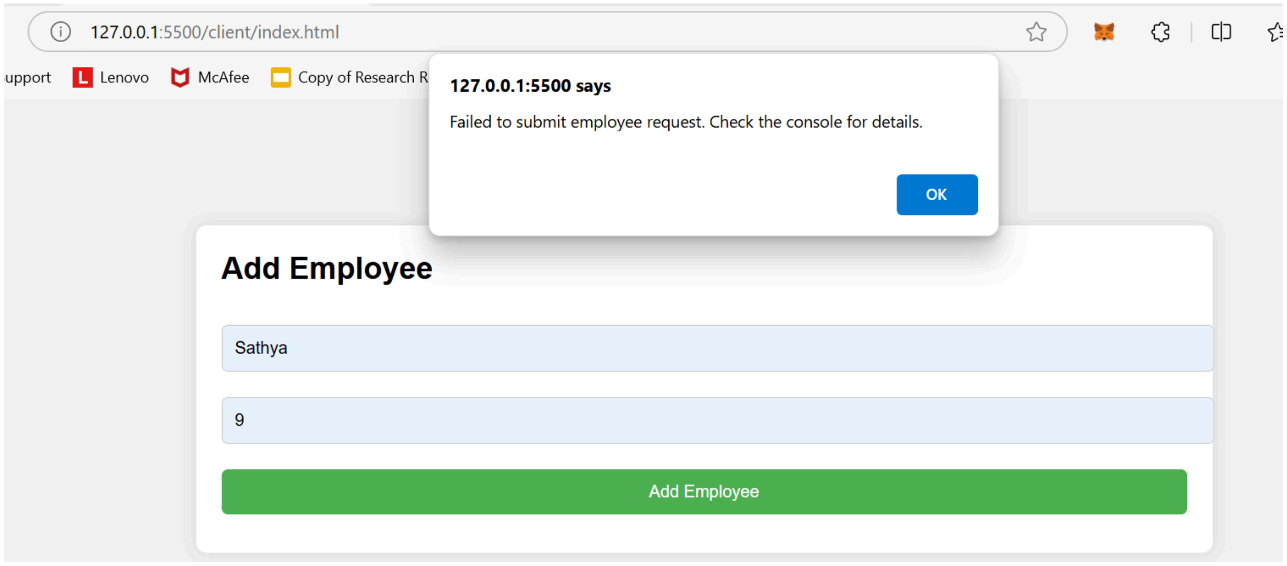
Add Employee

Sathya

9

Add Employee

Rejected



127.0.0.1:5500/client/index.html

upport Lenovo McAfee Copy of Research R

127.0.0.1:5500 says

Failed to submit employee request. Check the console for details.

OK

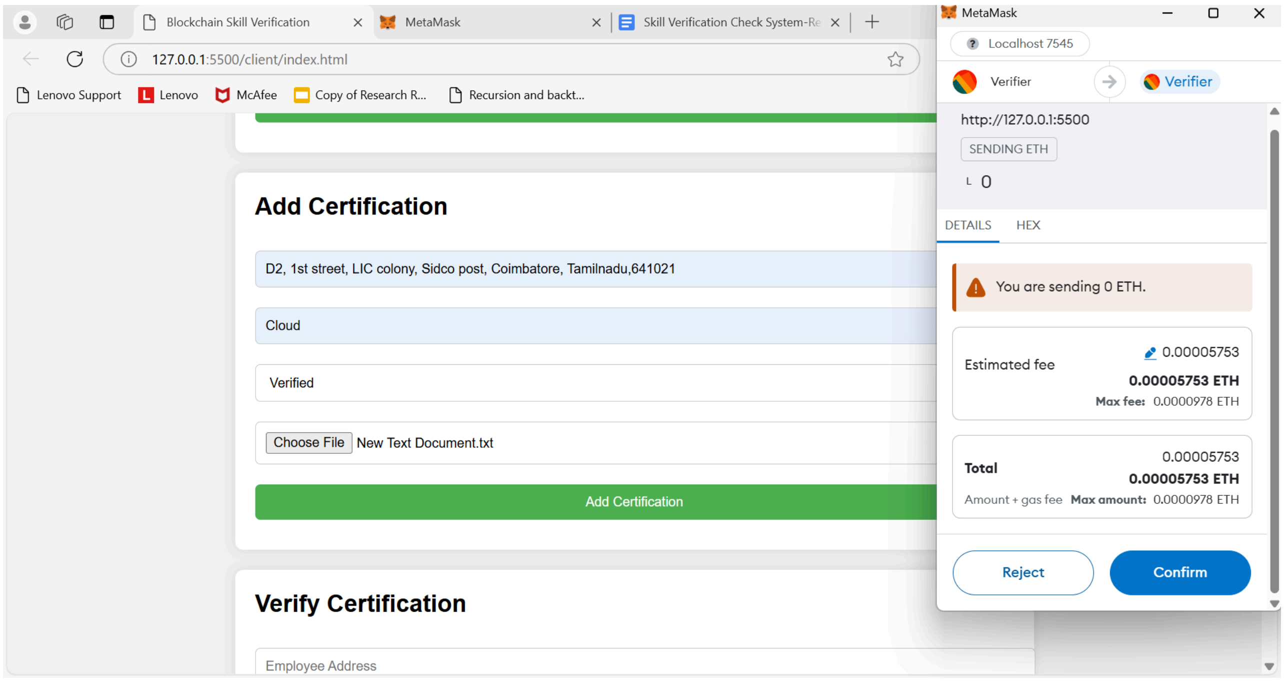
Add Employee

Sathya

9

Add Employee

Add Certification



Blockchain Skill Verification | MetaMask | Skill Verification Check System-Re

127.0.0.1:5500/client/index.html

Lenovo Support Lenovo McAfee Copy of Research R... Recursion and backt...

Add Certification

D2, 1st street, LIC colony, Sidco post, Coimbatore, Tamilnadu,641021

Cloud

Verified

Choose File New Text Document.txt

Add Certification

Verify Certification

Employee Address

MetaMask

Localhost 7545

Verifier → Verifier

http://127.0.0.1:5500

SENDING ETH

0

DETAILS HEX

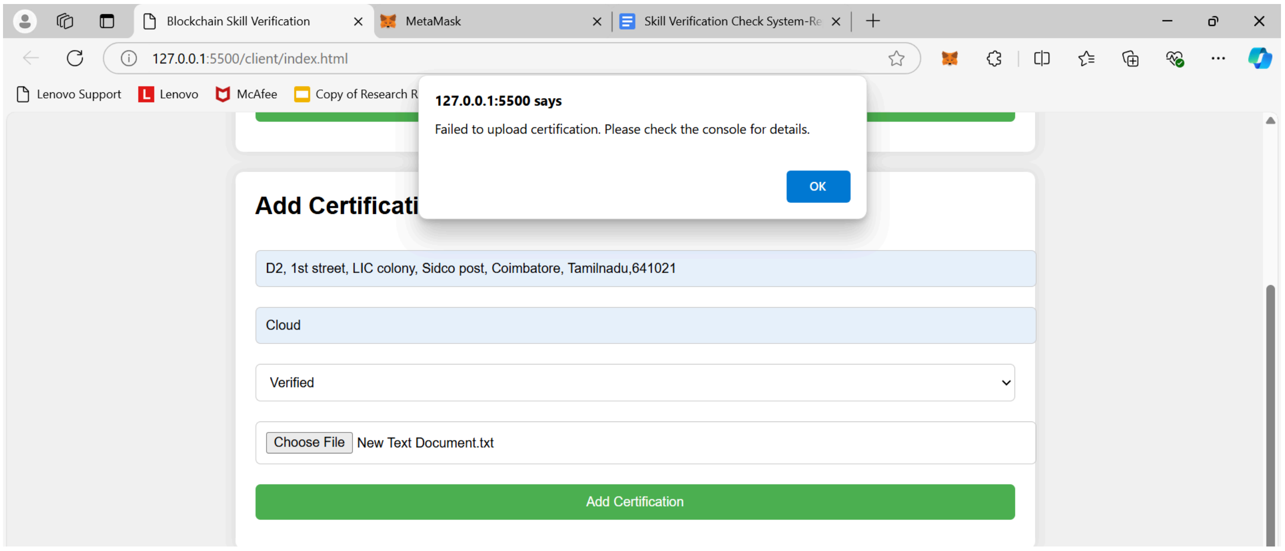
You are sending 0 ETH.

Estimated fee 0.00005753
0.00005753 ETH
Max fee: 0.0000978 ETH

Total 0.00005753
0.00005753 ETH
Amount + gas fee Max amount: 0.0000978 ETH

Reject Confirm

Rejected


Blockchain Skill Verification
MetaMask
Skill Verification Check System-Re
127.0.0.1:5500/client/index.html
Lenovo Support Lenovo McAfee Copy of Research R

127.0.0.1:5500 says

Failed to upload certification. Please check the console for details.

OK

Add Certificati

D2, 1st street, LIC colony, Sidco post, Coimbatore, Tamilnadu,641021

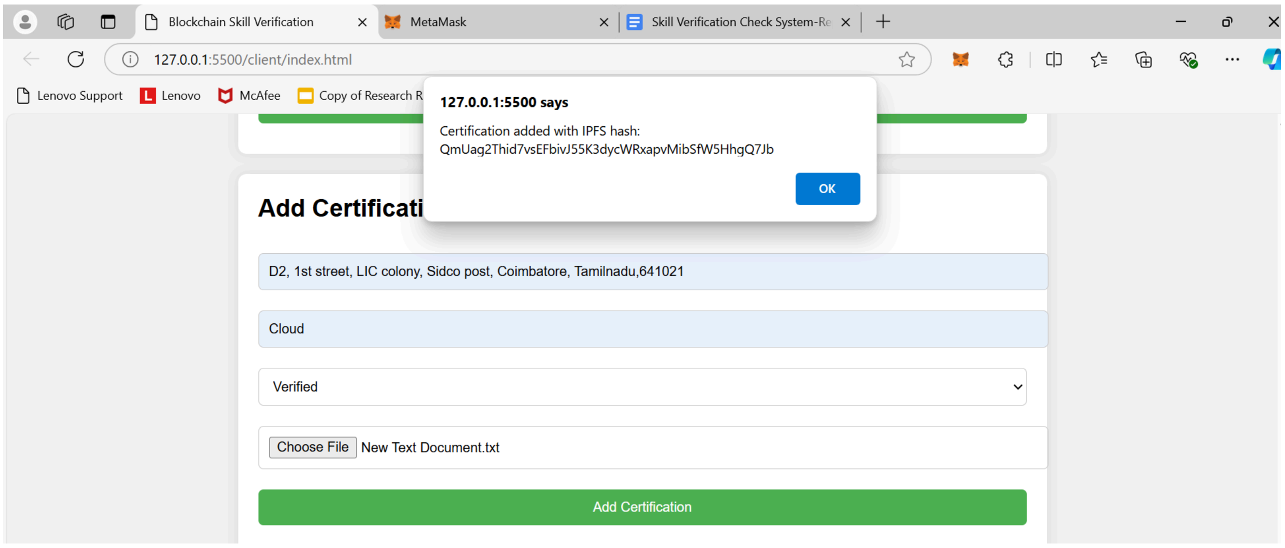
Cloud

Verified

Choose File New Text Document.txt

Add Certification

Accepted


Blockchain Skill Verification
MetaMask
Skill Verification Check System-Re
127.0.0.1:5500/client/index.html
Lenovo Support Lenovo McAfee Copy of Research R

127.0.0.1:5500 says

Certification added with IPFS hash:
QmUag2Thid7vsEFbivJ55K3dycWRxapvMibSfW5HhgQ7Jb

OK

Add Certificati

D2, 1st street, LIC colony, Sidco post, Coimbatore, Tamilnadu,641021

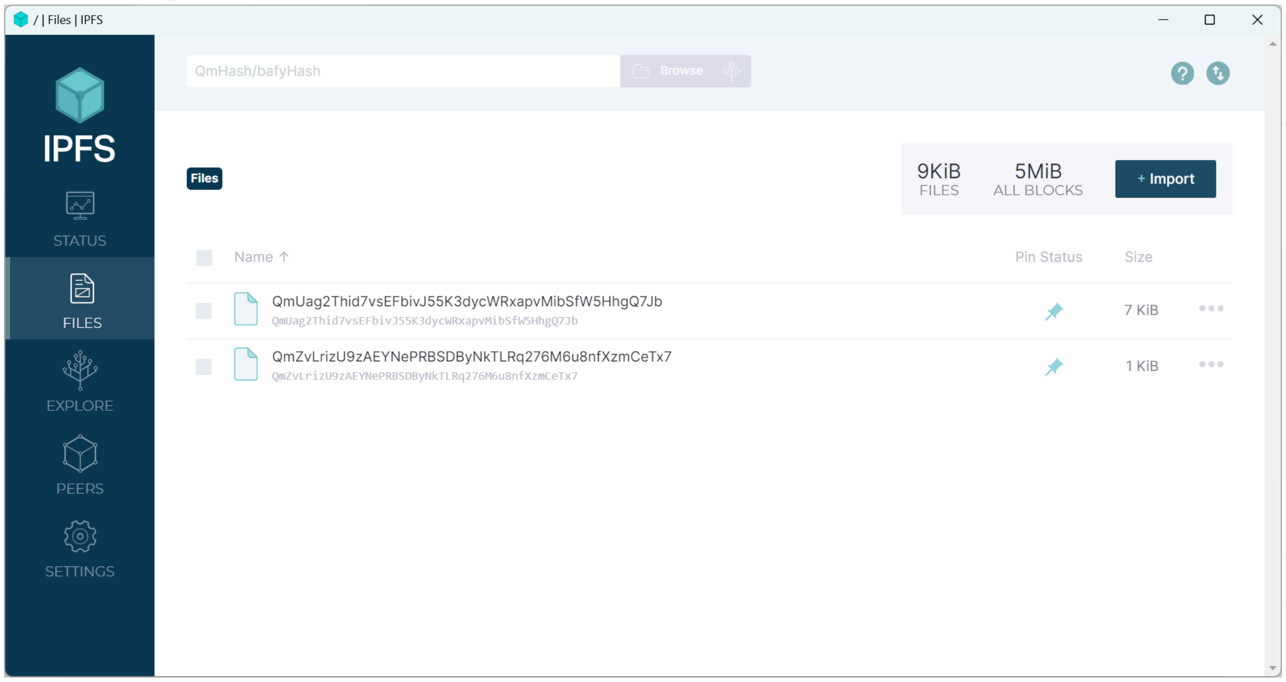
Cloud

Verified

Choose File New Text Document.txt

Add Certification

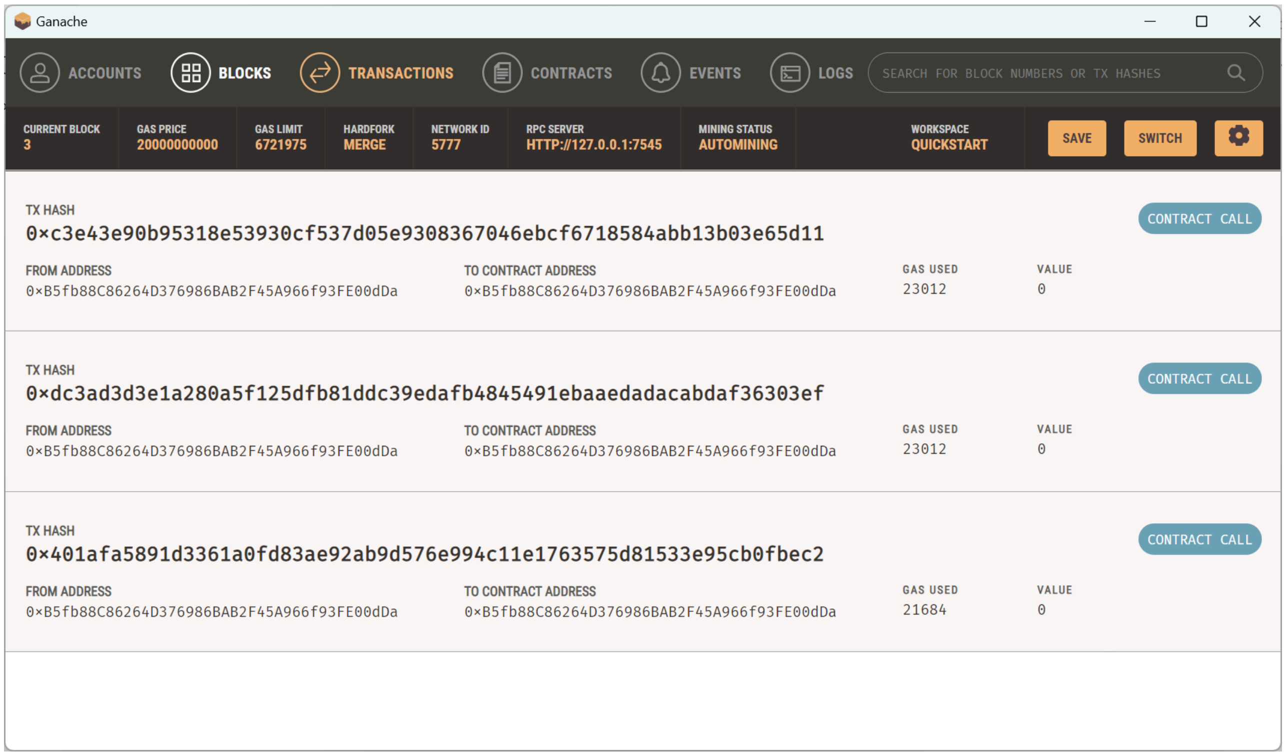
Document uploaded



9KiB FILES 5MiB ALL BLOCKS

Name ↑	Pin Status	Size
QmUag2Thid7vsEFbivJ55K3dycWRxapvMibSfW5HhgQ7Jb QmUag2Thid7vsEFbivJ55K3dycWRxapvMibSfW5HhgQ7Jb	📌	7 KiB
QmZvLrizU9zAEYNePRBSDByNkTLRq276M6u8nfXzmCeTx7 QmZvLrizU9zAEYNePRBSDByNkTLRq276M6u8nfXzmCeTx7	📌	1 KiB

Ganache transactions



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
3	20000000000	6721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART

TX HASH
0x c3e43e90b95318e53930cf537d05e9308367046ebcf6718584abb13b03e65d11
CONTRACT CALL

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	23012	0

TX HASH
0xdc3ad3d3e1a280a5f125dfb81ddc39edafb4845491ebaaedadacabdaf36303ef
CONTRACT CALL

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	23012	0

TX HASH
0x401afa5891d3361a0fd83ae92ab9d576e994c11e1763575d81533e95cb0fbec2
CONTRACT CALL

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	0xB5fb88C86264D376986BAB2F45A966f93FE00dDa	21684	0

CONCLUSION

Thus we have successfully developed the Skills Verification System that presents a modern solution to traditional credential verification challenges, promoting a more trustworthy and efficient hiring process. By harnessing cutting-edge technologies, it not only streamlines the management of employee qualifications but also reinforces the integrity of the hiring ecosystem. The implementation of this system can significantly benefit organizations by reducing fraud, increasing confidence in employee qualifications, and ultimately driving productivity in the workforce.

REFERENCES

1. Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum.org. https://ethereum.org/en/whitepaper/

2. Web3.py Documentation. (n.d.). Web3.py. https://web3py.readthedocs.io/

3. Truffle Suite Documentation. (n.d.). Truffle Suite. https://www.trufflesuite.com/docs

4. Solidity Documentation. (n.d.). Solidity Programming Language. https://docs.soliditylang.org/

5. MetaMask Documentation. (n.d.). MetaMask Docs. https://docs.metamask.io/

6. IPFS Documentation. (n.d.). IPFS Docs. https://docs.ipfs.io/

7. Grech, A., & Camilleri, A. F. (2017). Blockchain in Education. Publications Office of the European Union.

8. Tapscott, D., & Tapscott, A. (2017). The Blockchain Revolution and Higher Education. EDUCAUSE Review, 52(2), 10-24.

9. Gräther, W., Kolvenbach, S., Ruland, R., Schütte, J., Torres, C., & Wendland, F. (2018). Blockchain for Education: Lifelong Learning Passport. In Proceedings of 1st ERCIM Blockchain Workshop 2018. European Society for Socially Embedded Technologies (EUSSET).