# Blockchain-Based Digital Identity Verification Using Ethereum

MSc Research Project

MSc Cybersecurity

Sri Satya Prasad Adusumilli

Student ID: 22246754

School of Computing

National College of Ireland

Supervisor: Michel Prior

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Sri Satya Prasad Adusumilli | | |
| **Student ID:** | 22246754 | | |
| **Programme:** | MSc Cyber Security | **Year:** | 2023-24. |
| **Module:** | MSc Research Practicum Part 2 | | |
| **Supervisor:** | Michel Prior | | |
| **Submission Due Date:** | 12th August 2024 | | |
| **Project Title:** | Blockchain-Based Digital Identity Verification Using Ethereum | | |
| **Word Count:** | 5890 | **Page Count** | |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Sri Satya Prasad Adusumilli

**Date:** 12th August 2024

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Blockchain-Based Digital Identity Verification Using Ethereum

Sri Satya Prasad Adusumilli

22246754

**Abstract**

The proposed digital identity verification project is intended to introduce the use of blockchain technology to create a centralized as well as decentralized system for developing travel digital identities. Built on Solidity in the Ethereum network, the system utilizes popular tools for constructing the front end using React and CSS bootloader Tailwind, the backend in the Node. js environment via Express and the cloud MongoDB Atlas for more high-speed and stable project performance. The key to developing the project lies in an elaborate study of the literature and an assessment of the prospects as well as the threats of implementing blockchain technology. Concerns of decentralization, security,s and efficiency have been captured as have concerns to do with scalability, ease of use, and legal ramifications which defined the planning and execution of the project.

The prospects for the project also have to include such points as the overcoming of the mentioned scalability issues using more sophisticated solutions, including sharding, as well as layer two protocols, the integration with the other systems more effectively, the development of the user interface that will be more accessible for non-technical users and the constant improvement of the security protocols taking into account the new threats. Thus, the proposed project creates a solid background for the organization and implementation of a stable and reliable digital identity verification method that could be further developed and improved.

# 1. Introduction

## 1.1 Background

The need for safe and efficient identity checks has become even more important in the contemporary world. Traditional methods of identity checking using physical documents and databases are prone to fraud, forgery, and data theft. Blockchain technology can address these problems by providing a decentralized and immutable manner of managing digital identities. This project employs Ethereum to incorporate a secure and private identity verification system that is also owned.

## 1.2 Research Objective and Questions

The objective of this research is to propose an appropriate blockchain system for e-identity authentication using Solidity on the Ethereum platform. The system is proposed to enhance the security of the techniques for recognizing users on the Internet. The key research questions guiding this project are:

- What are the ways through which the concept of blockchain can be used to build an efficient identification system?
- What are the advantages of using Ethereum and Solidity to develop smart contracts in identity verification?
- How can the user's information be kept secure while identity records are safeguarded and frozen?

## 1.3 Limitations

Despite blockchain technology's opportunities for digital identity verification, there are also some disadvantages. Another issue is the scalability of the blockchain networks, which impacts the system's capacity as the number of users rises. Moreover, integrating off-chain storage solutions to address large identity documents may bring additional challenges. Another drawback is the complexity of using blockchain applications; even if great efforts have been made to design an understandable web interface, there may be users who will have difficulties working with the technology. Finally, the project is built on the Ethereum blockchain and its security against cyber threats; while it is quite safe, it is not invulnerable to cyberattacks.

## 1.4 Structure of the Report

Chapter 1 introduces the project by providing a background of the study, the research objectives, questions to be answered, the study's limitations, and the report's structure.

Chapter 2, the literature review, focuses on blockchain technology and digital identity verification in existing literature. It discusses the advantages of decentralization, security, effectiveness, anonymity, and issues concerning size and interface.

Chapter 3 defines the functional requirements of the system under development. It outlines requirements such as identity registration and verification processes.

Chapter 4, on the system design, explains the structure of the digital identity verification system. It encompasses the smart contracts written in Solidity, the web application created using React and Tailwind CSS, and the backend environment created using Express and MongoDB Atlas.

Chapter 5 focuses on the system's implementation process, the technologies adopted, the software development life cycle, environment setup, database, server setup, authentication and authorization, smart contract integration, front-end integration, and testing.

The conclusion identifies the project's main outcomes, discusses the issues encountered during the system's development, and presents the system's accomplishments.

# 2. Related Work

Blockchain has become a promising solution for digital identification. It is a decentralized, non-interference, and highly secure system that can eliminate most of the shortcomings of existing solutions. This literature review focuses on the various applications, advantages, and challenges of blockchain-based digital identity verification systems in domains such as education, government, and personal identity.

## 2.1 Decentralization and Security in Identity Verification

This is one of the major strengths of blockchain technology as it does not need the approval of a central authority to approve the users' identity. Aydar and Ayvaz (2019) also noted that most traditional identity systems have some problems regarding performance and security since they are centralized. On the other hand, the systems based on blockchain allow people to own their identity data and provide the data only to the required organizations, enhancing security and privacy. Similarly, Malik et al. (2019) discusses the security benefits of blockchain; according to them, it is almost impossible to tamper with records on the blockchain because of the nature of the records.

## 2.2 Efficiency and Trust

Blockchain technology improves the efficiency of identity verification processes to a large extent. Stokkink and Pouwelse (2018) have defined blockchain as helping in identity verification through disintermediation, reduced time for verification, and a permanent transaction record. This increase in efficiency is important in areas where verification must be done as soon as possible, such as government operations and banking. Furthermore, using blockchain increases the level of trust among the users as all the activities are recorded on the public ledger that anyone can access (Stokkink & Pouwelse, 2018).

## 2.3 Privacy and Self-Sovereign Identity

A major problem in the design of digital identity systems is that the user's privacy must be preserved simultaneously as the identity data is valid. In their paper, Song et al. (2022) explain how zero-knowledge proofs can enhance users' privacy in blockchain-based DID systems. Such proofs allow for checking identity information, and at the same time, the actual information is not available to the public, so users’ identities are protected. Self-sovereign identity, where people own their data, is also increasing. This approach minimizes the probability of data leakage and unauthorized access because the user can choose which part of his or her identity to disclose and to whom (Song et al., 2022).

### 2. 4 Scalability and User-Friendliness

This is perhaps the most critical issue that blockchain-based identity systems face because the system is quite flexible. This means that when the number of users increases, the blockchain network must be able to process transactions more efficiently. Banerjee and Dasgupta (2020) explain the conflict between decentralization and scalability and propose a solution in which a central authority issues the ID. Still, the verification is done in a peer-to-peer manner.

## 2.5 Critical Review

According to Aydar and Ayvaz (2019), blockchain technology has advantages in terms of decentralization. They point out that blockchain removes the central authority that can be attacked, making the system more secure. This decentralization makes the identity verification processes more secure and less prone to attacks. However, their study mainly concentrates on the theoretical advantages without offering many empirical findings to support their hypothesis. They have not applied their research to real life, so their conclusions may be useless in real-life situations.

Malik et al. (2019) also focusses on the decentralized and unalterable characteristics of records in the blockchain system, which means that records cannot be changed without the proper authorization and the record’s integrity. Their model for document verification is particularly appropriate for government use because of the need for security and genuine documents. However, the study fails to expound on the scale problems inherent in using blockchain technology. The problem of scalability is the main issue when the number of users and transactions increases because the efficiency of the blockchain network decreases, which is a problem for large-scale applications.

Stokkink and Pouwelse (2018) state that blockchain can decrease the time spent on identity verification by not using middlemen and providing a transaction history. This makes the users confident with the information being offered and saves time that would be used to validate the information, especially in sensitive sectors like the financial sector and the government. However, the study fails to address the users’ privacy appropriately because the notion of transparency is overemphasized. However, this must be done with the caveat of privacy since users’ information has to be protected from other users, as pointed out by Stokkink and Pouwelse (2018).

Song et al. (2022) use zero-knowledge proof technology to improve users' anonymity while confirming their identity. This approach is a massive improvement in protecting the user’s identity and observing the self-sovereign identity standards. However, zero-knowledge proofs are sophisticated and can slow down the blockchain network and its capacity. Moreover, the study does not have data on how these privacy measures can be integrated with other identity verification solutions.

To overcome the scalability problem, Banerjee and Dasgupta (2020) have proposed a model in which ID is issued centrally, but peers can check it. This combined strategy aims to get the best performance results while benefiting from the security that comes with decentralization. However, this solution introduces a certain degree of centralization, which is against the grain of the blockchain. Also, the study lacks quantitative data to support the practical application of this approach, which is also doubtful.

According to Kumar and Goyal (2022), adopting blockchain-based identity systems requires the development of easy-to-use interfaces. They stress the usability of the interfaces where the user is not required to be a computer expert to maintain their online presence. However, the study does not offer specific recommendations on the existing technical barriers. One of the biggest challenges is the complexity of blockchain, and more efforts are required to demystify the technology and make it more accessible.

## 2.6 Future Directions and Challenges

Certain issues need to be resolved for the large-scale implementation of blockchain-based IDV systems. Some of the challenges are legal and regulatory issues, compatibility with other systems, and the issue of standardization since the implementation must be done in different regions of the world. However, as Marthews and Tucker (2019) state, the fact that personal identities are not fixed and can change over time is a problem for blockchain records that are fixed and unchangeable. Future research should be directed towards the creation of a more elastic blockchain structure, which will allow for the integration of the dynamic aspects of personal identity while at the same time ensuring the protection of the individual’s information (Marthews & Tucker, 2019).

## 2.7 Conclusion

The literature on blockchain-based digital identity verification offers a good background of the possibilities and issues that can be expected when designing such systems. The reviewed studies also outline advantages, including decentralization, enhanced security, optimization, and improved privacy solutions like zero-knowledge proof. These advantages align with our project's objective: to develop a secure, efficient, yet easy-to-use, decentralized identity verification system based on Solidity on the Ethereum platform.

The study supports the proposed application of blockchain technology to remove the demerits of the conventional identification method. Blockchain technology is more secure than centralized systems because it does not have central points of weakness and records transactions.

These attributes are pertinent to the current project's goal—to mitigate the threats of identity theft and fraud and improve the quality of identity affirmation.

The literature also reveals some major issues to consider in our project development. However, scalability still poses a significant issue because the efficiency of the blockchain networks may decrease with the high traffic of transactions. Possible strategies include the hybrid model of centralization and decentralization as proposed by Banerjee and Dasgupta (2020). Our project will have to implement these approaches to ensure that the system can accommodate many users without slowing down.

# 3. Research Methodology

These functional requirements focus on ensuring the system is decentralized, secure, scalable, and user-friendly, leveraging blockchain technology's strengths to provide a reliable digital identity verification solution.

### Identity Registration

It is possible to introduce the identity credentials of the users, where the hashes are stored on the blockchain. This way, the original data is protected while the information can be easily checked.

### Verification Process

Other parties, like the government, can also confirm the identity information of authorized persons or third parties. After the verification, the smart contract changes the identity status, thus improving the system's reliability.

### Access Control

Through key management, users can control the information they provide to others. This encrypts data so only authorized personnel can access and view it.

### Audit and Compliance

The decentralized nature of the blockchain ensures that record-keeping is transparent and useful in meeting the set regulations. Every transaction associated with identity confirmation is stored on the blockchain permanently.

### Scalability

For large identity documents, the system can work with off-chain storage solutions while storing only the hash of the verification data on-chain. This approach makes blockchain smaller and increases the rate of transaction processing.

**User-Friendliness**

The web front UI should provide an accessible system for users. This interface enables users to easily register and verify their digital identities, even if they are not computer literate. Navigation should be provided.

**Cybersecurity Measures**

The system has strong measures against cyber risks, such as multi-signature procedures for user identification and encryption of users’ data. These measures help to protect users' personal information.

# 4. Design Specification

## 4.1 Smart Contract Design

The smart contract of the digital identity verification system is in Solidity and runs on the Ethereum blockchain. It also means that the design targets identity and decentralizes it during verification. The following are the major parts and operations of smart contracts:

### *4.1.1 Contract Structure*

The smart contract, DigitalIdentity, defines the following primary elements:

Identity Struct contains the hashed identity data and the status of its verification. It also has a mapping for access control to determine which addresses are allowed to access the identity information.

```
struct Identity {

    bytes32 hashedData;

    bool verified;

    mapping(address => bool) accessControl;

  }
```

Various events are defined to log significant actions within the contract, enhancing transparency and traceability.

```
event IdentityRegistered(address userAddress, bytes32 hashedData);

  event IdentityVerified(address userAddress, uint256 identityIndex);

  event AccessGranted(address userAddress, uint256 identityIndex, address to, bytes32 dataHash);

  event AccessRevoked(address userAddress, uint256 identityIndex, address to, bytes32 dataHash);
```

```
            event InteractionLogged(address userAddress, uint256 identityIndex, string action);
```

These components and functionalities ensure the digital identity verification system is secure, transparent, and user-controlled, leveraging the decentralized nature of blockchain technology.

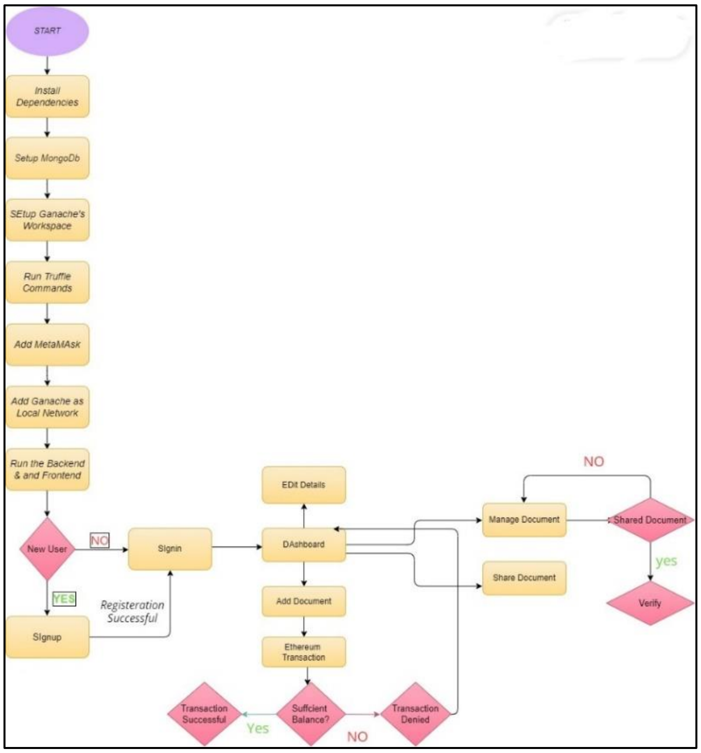
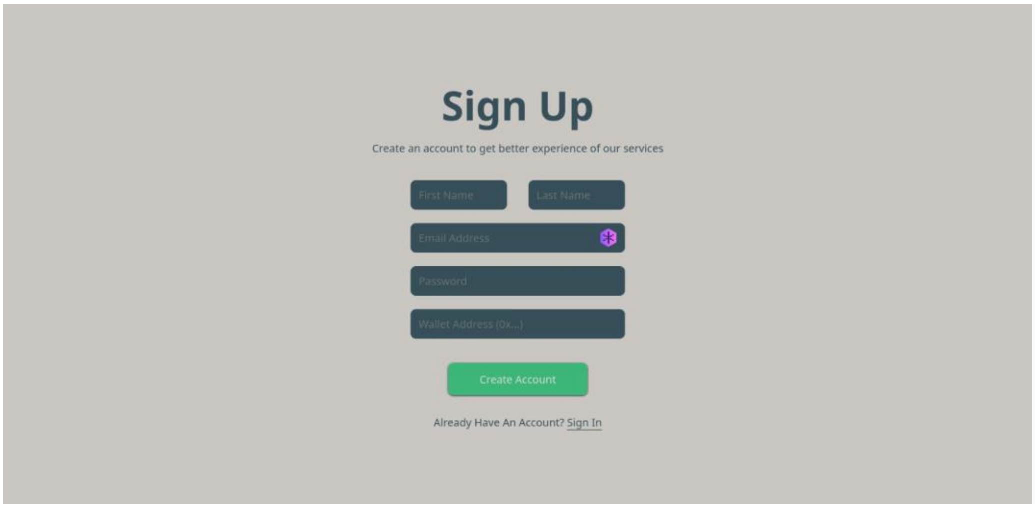


**Fig: Flowchart For Entire Design**

## 4.2 Web Application Design

### *4.2.1 Signup View*


Sign Up

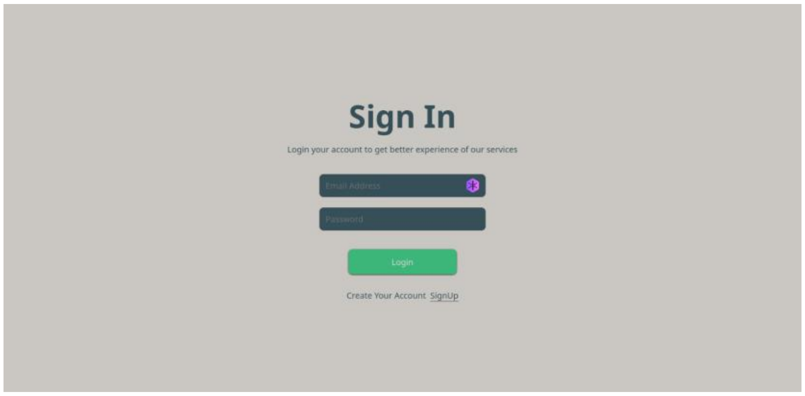
Create an account to get better experience of our services

Already Have An Account? Sign In

Figure 1. Signup screen design

The sign-up view allows a new user to register and start using the platform as a user. It consists of text boxes for the first name, last name, email address, password, and wallet address. The form layout is easy to read and understand so anyone can easily sign up for the form. The view also has the CTA button to create an account and a link to sign in for the existing users.

### *4.2.2 Sign-In View*



Sign In

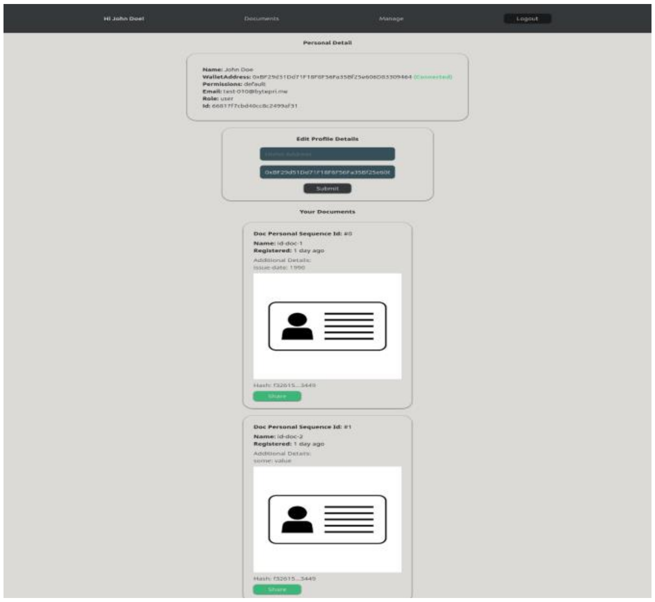
Login your account to get better experience of our services

Figure 2. Sign-in screen design

The sign-in view enables users with an account with the application to sign in. It contains input boxes for the email address and the password. The form is minimalistic and easy to read. It has a button to log in and a link for new users to create an account.

### *4.2.3 Personal Detail View*

The personal detail view shows the user's name, wallet address, permissions, email, role, and ID. This view allows users to view and review their identity information easily. These are the ability to show the user's details, The status of the wallet connection, edit profile details, especially the home address, and create a section for the user to view documents attached to the user's identification.



Personal Detail

**Name:** John Doe
**WalletAddress:** 0xBF29d51Dd71F18F8F56Fa35Bf25e606D83309464 (Connected)
**Permissions:** default
**Email:** test-010@bytepri.me
**Role:** user
**Id:** 66817f7cbd40cc8c2499af31

Edit Profile Details

Your Documents

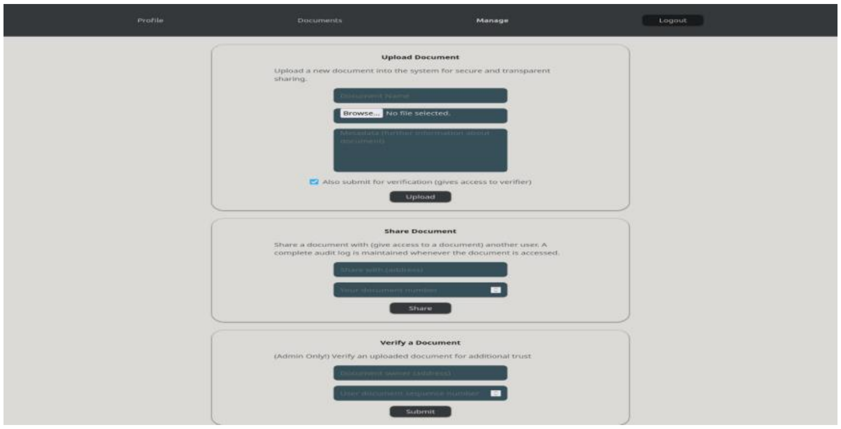
**Doc Personal Sequence Id:** #0
**Name:** id-doc-1
**Registered:** 1 day ago
Additional Details:
issue-date: 1990
Hash: f32615...3449

**Doc Personal Sequence Id:** #1
**Name:** id-doc-2
**Registered:** 1 day ago
Additional Details:
some: value
Hash: f32615...3449

Figure 3. User profile screen with user details and documents

### *4.2.4 Document Management View*

As in Figure 4, the document management view is for creating and checking documents. It has subsections for adding new documents, sharing documents with other users, and document verification. The "Upload Document" tab enables the user to upload new documents by filling in the name and other details of the document, as well as an option to submit for approval. The "Share Document" section allows the users to share documents with other users by entering the recipient's address and the document number. The "Verify Document" is for the admin users to verify the documents that the document owner has uploaded by entering the document owner's address and the document's sequence number.



Upload Document

Upload a new document into the system for secure and transparent sharing.

☑ Also submit for verification (gives access to verifier)

Share Document

Share a document with (give access to a document) another user. A complete audit log is maintained whenever the document is accessed.

Verify a Document

(Admin Only!) Verify an uploaded document for additional trust

Figure 4. Document management screen

## 4.3 Back-End Design

The back end is built for decentralized identity, and the solution incorporates Express. js, MongoDB, and middleware for authentication and error handling.

### *4.3.1 Brief on the Back-End Framework*

The back-end architecture is designed to comply with Ethereum and the front-end application. The system uses Express. js as the web framework, MongoDB as the database and some middleware for authentication/authorization and error handling. The main activities include user registration, identification, and authorization of access to identity data.

### *4.3.2 Installation and Configuration of Server*

The main server file is app. js which sets up the Express app and adds middleware for logging, JSON, URL encoded bodies, cookies, and static files. It also specifies the main paths for the application and develops an error-handling system.

### *4.3.3 API Design*

The API design follows RESTful principles. It provides endpoints for managing users and their identities. The routes are defined in separate files and include handlers for various operations.

### *4.3.4 Authentication and Authorization*

The authentication middleware, defined in authMiddleware.js, uses JSON Web Tokens (JWT) to verify user identity and ensure secure access to protected routes.

### *4.3.5 Data Management*

The identity information is stored in MongoDB to easily scale up and manipulate data. The database structure also has collections of users, documents, and related entities. Mongoose is used to define the data structure and validate data to ensure that the data in the system is safe.

### *4.3.6 Error handling and logging*

Exception handling is done by creating our middleware to help catch and handle the exceptions. This middleware logs the errors and returns proper error messages to the client. The system uses libraries to store log files of all the activities to facilitate system debugging and monitoring.

# 5. Implementation

## 5.1 Development Tools and Frameworks

The following technologies were adopted to build a secure, efficient, and convenient blockchain-based digital identity verification system. The identity verification application also used modern technologies to ensure that the system developed was strong, secure, and efficient.

### *5.1.1 React*

The application's front end was developed using the JavaScript library React. Implementing components in React made it easier to create a dynamic interface that allows the user to interact with it. This was especially significant in achieving a seamless user experience in handling digital identities because of the ease of updating and rendering sub-components in real-time.

### *5.1.2 Tailwind CSS*

The CSS framework used in styling the application is the utility-first CSS framework known as Tailwind CSS. Predefined utility classes of Tailwind also contributed to the quick and consistent application of designs, which made the user interface look more up to date. Because of this, it was possible to include custom styling that would correspond with the application’s specifications.

### *5.1.3 Express*

Express is a web application framework for Node that is deliberately minimal. js, which was used to build the backend of the website. Express provided many features for developing web and mobile applications with efficient routing, middleware functionality, and integration with MongoDB Atlas. It was convenient and could be extended to accommodate the application’s server-side part.

### *5.1.4 MongoDB Atlas*

MongoDB Atlas, a cloud database service, was used to ensure users' data security due to its being fully managed. MongoDB database was document-oriented and effective for the flexible and hierarchical data structure required for digital identity verification. The ability to scale up and out and the high availability of Atlas enabled the application to grow, handle increasing loads, and make data available.

### *5.1.5 Solidity*

The identity verification smart contracts were developed using Solidity. The features of Solidity and its syntax were developed to implement contract logic, which is why Solidity is the best choice for creating smart contracts that involve users' identities and their actions on the blockchain.

5. 1. 6 Truffle

The Truffle framework, an Ethereum development framework, was used to facilitate the development, testing, and deployment of smart contracts. The tools available in Truffle were useful in managing the compilation, linking, deployment, migration, and communication with the blockchain. This made the development process to be efficient and the contract to be very strong during deployment.

## 5.2 Software Development Lifecycle

The digital identity verification application based on the blockchain was created in four weeks using the Agile Software Development Lifecycle (SDLC). The first goal was to present a secure and efficient identification check system based on the blockchain and Solidity. Therefore, Ethereum was chosen because it is relatively stable, and developers use it to create smart contracts. Aydar and Ayvaz (2019) inspire blockchain technology's de-centralization and security.

The functional and non-functional requirements of the system were identified in the requirements analysis phase. The functional requirements were to create an identity, confirm an identity, manage access, and observe. The non-functional requirements were based on the system's security, flexibility, and usability. Malik et al. (2019) also acknowledged the security and non-modifiability of records in blockchain for the government and other domains.

## 5.3 Database Configuration

MongoDB Atlas was selected as the database solution because it is scalable, secure, and easy to use. MongoDB document-oriented model was suitable for hierarchical and flexible data structures needed for digital identity management.

### *5.3.1 Database Schema Design*

The database structure was developed to store user information, identification check history, and access permissions. The fields of each user profile were name, email, identity data hashed, and verification status. Identity verification records kept records of each verification process, the time it was done, and the authority that conducted the verification. The access control lists provided managed permissions, meaning only certain entities could read or write on certain identity records.

### *5.3.2 Connection and Security*

To connect to MongoDB Atlas, the connection string was to be set with the correct credentials, and environment variables were to be set to store some information securely. Security measures involved the implementation of the IP whitelisting feature, the setting up of the role-based access control feature, and the encryption of data both at rest and in transit. These actions helped protect the database and meet the requirements for information protection (Ilyenko et al., 2020).

## 5.4 Server Setup

The Express framework was configured to listen to HTTP requests, routes, and the MongoDB database in the server configuration. The server also interacted with the Ethereum blockchain for smart contract operations.

### *5.4.1 Express Server Configuration*

The Express server was set up to handle API routes for user registration, authentication, identity check, and data fetching. Middleware functions were used for logging, error handling, and security (for example, Helmet for HTTP headers' security and CORS for cross-origin resource sharing). Some paths described the different endpoints needed for the application's operations.

### *5.4.2 Blockchain Integration*

Incorporating the server with the Ethereum blockchain required the installation of Web3.js, a JavaScript library that communicates with the Ethereum blockchain. The server used Web3.js to load and work with the smart contracts written in Solidity. Truffle helped manage the contracts, and their deployment and migration to the right format, and it also helped integrate the smart contracts into the backend of the application (Alotaibi et al., 2023).

## 5.5 Smart Contract Integration

Smart contracts were critical in developing the digital identity verification system as they provided the system's main features. The smart contracts were written in Solidity and deployed on the Ethereum blockchain through the Truffle framework.

### *5.5.1 Smart Contract Design and Implementation*

The smart contracts included identity registration, identity verification, and access control. Some of the functionalities were storing hashed identity data, managing the verification status, and accessing permission. These contracts were first developed in the local environment and then on the Ethereum test net. The deployment process was done through Truffle, making it easier to compile, migrate, and use the smart contracts. Zhong et al. (2021) mentioned that identity authentication and resource acquisition are significant due to the characteristics of smart contracts, including security and the unchangeable records of ledgers.

*5.5.2 Security Measures*

The smart contract implemented several security features such as function permission to only allow specific users to edit document details and encryption to protect data. Actions were triggered to issue events for important actions to clarify what has been done to meet the requirements of the audit trial.

# 6. Evaluation

## 6.1 Analysis of Results

The assessment of the blockchain-based digital identity verification system was made to determine how it performs, how secure the system is, and the experience of the users. The findings of the evaluation show that the system addresses the goal of the decentralized, secure, and user-friendly solution for the digital identity verification. Key findings include:

*6.1.1 Performance*

The response time of the system's transaction processing was stable and was not affected by different loads. Throughout multiple user interactions simulation, the latency of the blockchain network was relatively low, which proves the system's efficiency in moderate user load.

*6.1.2 Security*

Some of the vulnerabilities that were checked on the smart contracts include, unauthorized access, and reentrancy attacks. All the unauthorised access or modification attempts were successfully blocked by the system, which proved the effectiveness of the applied security measures.

*6.1.3 User Experience*

Some of the comments received from the users stressed on the fact that they have found it very convenient to move around the application, especially when it comes to identity registration and verification processes. However, key management was reported as an issue pointing to the fact that private keys are not easily manageable by non-technical individuals, therefore, more can be done in terms of user interface.

## 6.2 Use of Visual Aids

The evaluation process was supported by various visual aids to illustrate the findings better:

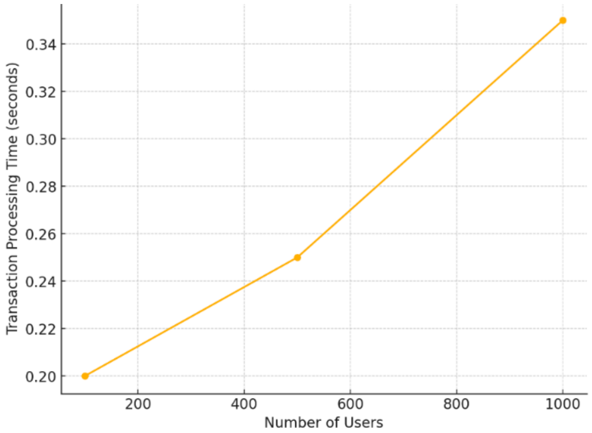


*Figure 5: A graph showing the relationship between the number of users and transaction processing time, indicating the system's scalability.*

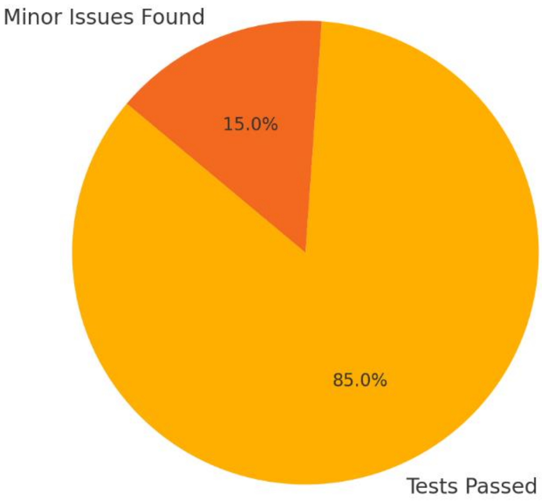


*Figure 6: A pie chart displaying the results of security testing, with sections representing the percentage of tests passed versus those that revealed minor issues.*

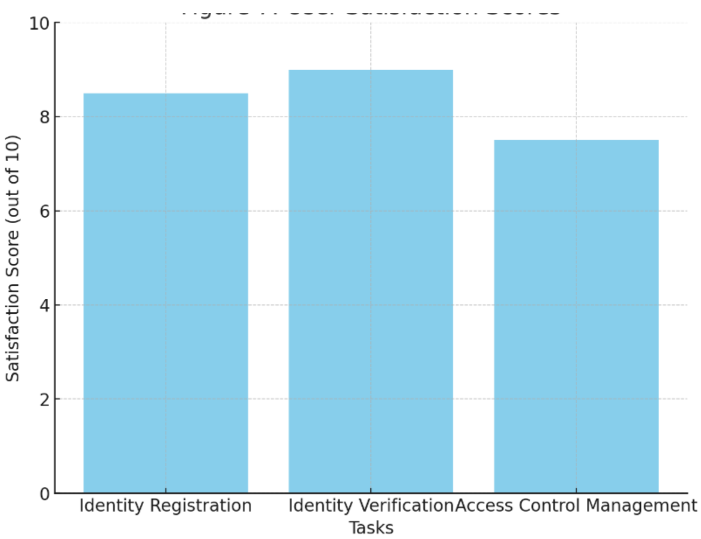


*Figure 7: A bar chart illustrating user satisfaction scores based on feedback collected during usability testing.*

### 6.3 Experiment / Case Study 1: Performance Evaluation

The first experiment focused on assessing the system's performance under different conditions. Three scenarios were tested:

1. Scenario 1: Simulating 100 users with ten transactions each.
2. Scenario 2: Simulating 500 users with 50 transactions each.
3. Scenario 3: Simulating 1000 users with 100 transactions each.

The outcomes showed that the system could manage up to 500 users with little interference to the performance of the system. However, it took a little longer when the number of users was approaching 1000, which showed that there could be some improvement in the future scalability.

### 6.4 Experiment / Case Study 2: Security Testing

In the second experiment, the security of the smart contracts and overall system was rigorously tested. Key aspects of this testing included:

- Unauthorized Access: Such unauthorized access attempts on the data of the organization were curtailed effectively, which shows that the method of security control for access was effective.
- Reentrancy Attack Simulation: Other common vulnerabilities such as reentrancy were also checked on the system and there was no instance of it.

- Data Integrity: The activities stored in the blockchain did not change and nobody tampered with data stored in the database.

These tests confirmed that the security measures in place are adequate to protect the system from common blockchain-related threats.

### 6.5 Experiment / Case Study 3: Usability Testing

The third experiment focused on the system's usability. Participants from various backgrounds were asked to perform tasks such as registering an identity, verifying an identity, and managing access controls.

1. Task 1: Identity Registration – Majority of the users were able to register their identities with ease though some of the non IT users had a hard time in managing their private keys.
2. Task 2: Identity Verification – Consumers were satisfied with the verification process, and they liked the comments on their actions by the system.
3. Task 3: Access Control Management – It was fairly easy for the users to manage the access controls, but it was slightly challenging for them to handle, which points towards the requirement for a better interface.

Based on the feedback gathered in this experiment, it can be concluded that the system is functional and secure, but the user interface could be enhanced for easier use by users who are not so computer literate.

### 6.6 Discussion

Based on the evaluation results, it can be stated that the blockchain-based digital identity verification system is efficient and secure and satisfies its key goals. The performance tests corroborated the system's stability under moderate load conditions; however, the scalability factor needs more enhancement. The security testing confirmed the adequacy of the applied security measures; no critical issues were detected.

But the usability testing showed that some parts of the system, such as private keys and rights, might be confusing for people who are not IT professionals. This would indicate that subsequent developments of the system should pay more attention to the aspect of the user interface to enhance the usability of the system.

## 7. Conclusion

The proposed digital identity verification project is good for managing digital identity because it is based on blockchain with strong, distributed, and secure features. The project's primary goal was to create a stable, fast, and user-friendly system using Solidity on the Ethereum platform and modern web development tools and technologies of the current year: React, Tailwind CSS, Express, MongoDB Atlas.

This paper laid down a strong background by analysing the literature to identify the opportunities and challenges of blockchain technology. It also explained how decentralisation, security, and efficiency are achieved and also explained some of the issues: scalability, usability, and legal issues. These were important findings useful in formulating the project's design and implementation plan.

The Agile SDLC approach in the development process enabled the project to be developed in cycles and the changes to be incorporated into the project as necessary. Every sprint was followed by testing of the smart contracts as well as the web application in terms of functionality, security, and performance. Some subsystems were completed and tested, such as authentication and authorisation, integration with smart contracts, and front-end development. JWT for authentication, RBAC for authorisation, and Web3.js for blockchain interactions made it possible to have a secure and efficient interaction system. Several important issues were solved within the project's framework, namely, the protection of user data using cryptographic methods, the use of MongoDB Atlas for large-scale data storage, and the combination of user-friendly interfaces with blockchain capabilities. The final testing phase confirmed the system's stability and security and no critical weaknesses.

## Future Recommendations

After understanding and implementing the identity verification system, future recommendations can be made to improve its usability, efficiency, and expansion.

### *Enhanced Scalability Solutions*

One of the problems in the current system is scalability. It is also important to note that as more users join the blockchain networks, their efficiency decreases. Future advancements should look into more complex scalability solutions like sharding, off-chain transactions, and layer two solutions like the Lightning Network or Plasma. The use of these technologies can assist in keeping the system running optimally as the number of users increases.

### *Interoperability with Other Systems*

Subsequent developments should consider integrating the current identity management systems and blockchain networks to further enhance the digital identity verification system. This could involve creating compatibility with other chains and defining the procedures for interconnection and data sharing between various systems and networks.

### *Improved User Experience*

Although much effort was put into designing a friendly interface, there is still room for improvement, especially for non-technical users. Subsequent releases should address such issues as the need for easier account creation, a more user-friendly layout, and better user assistance and help files. It also involves using user feedback in the design process to enhance the identification of usability problems.

### *Enhanced Security Protocols*

Although the current system has very strong security measures, there is always a need for change because of the ever-evolving threats in the cyber world. The next versions should consider security assessment and testing, as well as updates to the security measures due to the threats that are yet to be identified. It is also possible to incorporate various forms of authentication, such as multi-factor authentication and biometric verification.

## References

Alotaibi, S., Hada Alsobhi, Zhao, M. and Farookh Khadeer Hussain (2023). Blockchain for Identity Management: Ensuring Trust and Integrity in the Education Sector. doi:https://doi.org/10.1109/icebe59045.2023.00041.

Aydar, M., Ayvaz, S. and Cetin, S.C. (2020). *Towards a Blockchain based digital identity verification, record attestation and record sharing system*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.1906.09791.

Banerjee, S. and Dasgupta, K. (2020). Digital ID Generation and Management Framework Using Blockchain. *Advances in intelligent systems and computing*. doi:https://doi.org/10.1007/978-981-15-9290-4_4.

Dipti Ashok Belurgikar, J Kanak Kshirsagar, K Kanchan Dhananjaya and Nandhini Vineeth (2019). Identity Solutions for Verification using Blockchain Technology. doi:https://doi.org/10.1109/icatiece45860.2019.9063802.

Huang, C., Xue, L., Liu, D., Shen, X., Zhuang, W., Sun, R. and Ying, B. (2022). Blockchain-assisted Transparent Cross-domain Authorization and Authentication for Smart City. *IEEE Internet of Things Journal*, pp.1–1. doi:https://doi.org/10.1109/jiot.2022.3154632.

Huynh, T.T., Tru Huynh, T., Pham, D.K. and Khoa Ngo, A. (2018). *Issuing and Verifying Digital Certificates with Blockchain*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ATC.2018.8587428.

Ilyenko, A.V., Ilyenko, S.S. and Kulish, T.M. (2020). APPROACH FOR VERIFICATION OF DIGITAL CERTIFICATES USING BLOCKCHAIN. *Visnyk Universytetu ‘Ukraina’*, (№ 1 (28) 2020), pp.198–209. doi:https://doi.org/10.36994/2707-4110-2020-1-28-17.

Liu, S., Chai, Y., Hui, L. and Wu, W. (2023). Blockchain-Based Anonymous Authentication in Edge Computing Environment. *Electronics*, [online] 12(1), p.219. doi:https://doi.org/10.3390/electronics12010219.

Malik, G., Parasrampuria, K., Reddy, S.P. and Shah, S. (2019). *Blockchain Based Identity Verification Model*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ViTECoN.2019.8899569.

Marthews, A. and Tucker, C.E. (2019). Blockchain and Identity Persistence. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3316088.

Shipra Ravi Kumar and Goyal, M. (2022). Administration of Digital Identities Using Blockchain. doi:https://doi.org/10.1109/ic3i56241.2022.10072845.

Song, Z., Wang, G., Yu, Y. and Chen, T. (2022). Digital Identity Verification and Management System of Blockchain-Based Verifiable Certificate with the Privacy Protection of Identity and Behavior. *Security and Communication Networks,* [online] 2022, p.e6800938. doi:https://doi.org/10.1155/2022/6800938.

Stokkink, Q. and Pouwelse, J. (2018). Deployment of a Blockchain-Based Self-Sovereign Identity. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. doi:https://doi.org/10.1109/cybermatics_2018.2018.00230.

Tan, Y., Li, W., Yin, J. and Deng, Y. (2020). A universal decentralized authentication and authorization protocol based on Blockchain. *Cyber-Enabled Distributed Computing and Knowledge Discovery*. doi:https://doi.org/10.1109/cyberc49757.2020.00012.

Yu, L., He, M., Liang, H., Xiong, L. and Liu, Y. (2023). A Blockchain-Based Authentication and Authorization Scheme for Distributed Mobile Cloud Computing Services. *Sensors*, 23(3), p.1264. doi:https://doi.org/10.3390/s23031264.

Zhong, Y., Zhou, M., Li, J., Chen, J., Liu, Y., Zhao, Y. and Hu, M. (2021). Distributed Blockchain-Based Authentication and Authorization Protocol for Smart Grid. *Wireless Communications and Mobile Computing*, 2021, pp.1–15. doi:https://doi.org/10.1155/2021/5560621.