

BLOCKCHAIN IDENTIFICATION CONTROL: SPARKING A DATA PROTECTION REVOLUTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

PALLETI KIRAN KUMAR REDDY (Reg.No - 40110889)
MANIKINDI VISHNU KUMAR (Reg.No - 40110727)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL - 2024

BLOCKCHAIN IDENTIFICATION CONTROL: SPARKING A DATA PROTECTION REVOLUTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

PALLETI KIRAN KUMAR REDDY (Reg.No - 40110889)
MANIKINDI VISHNU KUMAR (Reg.No - 40110727)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PALLETI KIRAN KUMAR REDDY (40110889)** and **MANIKINDI VISHNU KUMAR (40110727)** who carried out the Project entitled "**Blockchain Identification Control: Sparking a Data Protection Revolution**" under my supervision from November 2023 to April 2024.

Internal Guide

Dr. JEBERSON RETNA RAJ MTech., Ph.D.,

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Head of the Department, Gandhi Salai,
Chennai - 600 119.

Dr. L. LAKSHMANAN M.E., Ph.D.,

Submitted for Viva Voce Examination held on 15/4/24

Internal Examiner

External Examiner

DECLARATION

I, Palleti Kiran Kumar Reddy hereby declare that the Project Report entitled "Blockchain Identification Control: Sparking A Data Protection Revolution" done by me under the guidance of Dr.R. Joberson Retna Raj, MTech., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 15/4/24

Kiran Kumar Reddy

PLACE: Chennai

P. Kiran Kumar
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, and **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R. Jeberson Retna Raj, MTech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Identity theft is the unauthorized acquisition of another person's confidential information in order to misuse it. Organizations and individuals should exercise caution when it comes to protecting their identities in order to avoid fraud as a result of identity theft. This information is freely available to attackers in user profiles. Attackers use this information to obtain additional information without raising suspicions about a final attack, identity theft, or fraud. Our block chain based Personal Identity Security System assists in securely storing personal identity data without fear of it being compromised or lost. In this system, the admin can access all the users and verify those accounts. The admin has access to all of the user's documents that have been uploaded to the system. They can view all of the user's activity logs. The admin can view the status and see if any fabrication is taking place. The admin can view the user's complaints. The user can manage their documents in the system by adding, updating, and deleting them. All documents uploaded by the user are also verified by the admin. The user can view the QR Code as well as the status of the documents that are being fabricated. The user can view if logs have been updated or deleted, as well as the status of fabricated document details. If a user discovers a fabrication or an action that they did not perform, they can notify the admin. The user can view all of the details by scanning the QR Code. They can add and verify all the documents with the ledger.

TABLE OF CONTENTS

Chapter No:	TITLE	Pg No:
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
	INTRODUCTION	
1	1.1 Introduction	1
	1.2 Problem Statement	2
	1.3 Objective and Scope	2
	1.4 Application of Blockchain IDMS	3
	LITERATUTE SURVEY	
2	2.1 Inferences from Literature Survey	5
	2.2 Existing System	5
	2.3 Problems with Existing System	6
	REQUIREMENT ANALYSIS	
3	3.1 Functional Requirements	9
	3.2 Non-Function Requirements	10
	3.3 Software and Hardware Requirements	12
	3.4 Java	12
	3.5 SQL (Structured Query Language)	14
	3.6 HTML (Hypertext Markup language)	15
	3.7 CSS (Cascading Style Sheet)	16
	3.8 Java Script	16
	3.9 Bootstrap	17
	3.10 Libraries and Classes Used	18
	DESCRIPTION OF PROPOSED SYSTEM	
4	4.1 Selected Methodology or Model	19
	4.2 Description of Software for Implementation	21
	4.3 System Architecture	22

	4.4 Workflow for Blockchain IDMS	22
	4.5 Modules	25
	4.6 Algorithms and Mechanisms Used	30
5	IMPLEMENTATION DETAILS	
	5.1 Implementation	33
6	RESULTS AND DISCUSSIONS	35
7	CONCLUSION	36
	REFERENCES	37
	APPENDIX	
	A. SOURCE CODE	39
	B. SCREENSHOTS	55
	C. RESEARCH PAPER	59
	D. CERTIFICATE	66

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Architecture Diagram	22
4.2	Work Flow Diagram	23
4.3	Admin Login Page	25
4.4	User Login Page	28
4.5	User Module	29
4.6	POA Consensus Mechanism	30
4.7	SHA-256 Block Diagram	31
4.8	AES Algorithm Working	32
4.9	File Request Count	34
B.1	Home Page	54
B.2	Admin Login Page	54
B.3	Admin Home Page	55
B.4	User Login Page	55
B.5	User Home Page	56
B.6	View Attribute Details	56
B.7	File Request Count	57
B.8	Qr Code Generation Page	57

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
SHA	Secure Hash Algorithm
AES	Advanced Encryption Standard
IDMS	Identity Management System
KYC	Know Your Customer
POA	Proof Of Authority

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Data leaks or identity fraud (e.g., stealing of credentials or credit card information) frequently affect profitable loss and drop of trust in the identity providers. Within this environment, new identity operation results that satisfy the factual requirements must be explored. The particular data is traditionally stored in a centralized system, which makes it possible for hackers or bushwhackers to achieve their vicious pretensions by robbery/ misusing/ manipulating these data in this centralized system. To gain access to the services handed by these online operations, the customers must validate their identity by giving a valid identifier. This identifier can take numerous forms similar to particular information contained on a social security card, automobile license, passport, birth certification, and academy or work badges. Identity Management is the process of authenticating, authorizing, and identifying a thing (individually or a group of people or the system) to access one or further many resources. The biggest point of the blockchain is its decentralization where the whole database is maintained by all the bumps on the network. An agreement medium ensures that the creation and modification of data are agreed upon by all the bumps or the maturity of the bumps. Using this fashion, the blockchain has the features of high security, not easy to be tampered or altered with, and so on. By decentralizing and encrypting identity data across a distributed ledger, blockchain promises to create a new paradigm where individuals have greater control over their personal information while drastically reducing the risk of unauthorized access and fraudulent activities. This project aims to delve into the intricacies of blockchain-based identity management, exploring its capabilities, benefits, and the challenges it presents, with a vision to illuminate the path towards a more secure and user-centric digital future.

Blockchain technology makes information about identity verifiable and auditable in seconds. Before diving into how blockchain is used in identity management, it's important to first understand the key features of blockchain and how it works.

With blockchain identity solutions, users don't need to worry about third-parties being involved in the verification processes or providing sensitive information to anyone else but the verifier. This ensures greater protection from fraud and identity theft since all user data is cryptographically protected and only stored by the users in their identity wallet apps.

1.2 PROBLEM STATEMENT

The current landscape of identity management systems faces significant challenges, including security vulnerabilities, privacy concerns, and inefficiencies. Centralized systems store user data in single, vulnerable locations, making them targets for cyberattacks and exploitation. Moreover, users often lack control over their personal information, leading to privacy breaches and data misuse. To address these issues, the development of a blockchain-based identity management system is imperative. By leveraging the decentralized and immutable nature of blockchain technology, such a system can offer enhanced security, privacy, and user control. It would utilize encryption, consensus mechanisms, and cryptographic techniques to ensure the integrity of user data while empowering individuals to selectively disclose information as needed. Interoperability with existing frameworks, scalability to accommodate growing demands, and compliance with regulatory requirements are essential considerations in the system's design. Additionally, user-friendly interfaces and transparent governance models are crucial for promoting adoption and trust among individuals, businesses, and organizations. Overall, a blockchain identity management system has the potential to revolutionize identity verification processes, offering a more secure, efficient, and user-centric solution to the challenges faced by traditional systems.

1.3 OBJECTIVE AND SCOPE

Objective

The objective of implementing a blockchain identity management system is to ignite a data security revolution. By harnessing the power of blockchain technology, this system aims to fundamentally transform how identity information

is managed, stored, and utilized. At its core, the objective is to establish a decentralized and immutable infrastructure that significantly enhances the security, privacy, and integrity of user data. Through cryptographic techniques and consensus mechanisms, the system ensures that sensitive information remains tamper-proof and resistant to unauthorized access or manipulation. Moreover, by empowering individuals with greater control over their personal data and enabling selective disclosure, the system fosters a paradigm shift towards user-centric identity management. This revolution not only addresses the shortcomings of centralized systems, such as susceptibility to cyberattacks and privacy breaches but also paves the way for a more transparent, efficient, and trustworthy ecosystem for identity verification. Ultimately, the goal is to establish a new standard of data security that not only mitigates risks but also catalyzes innovation and empowerment in the digital age.

Scope

The scope of a blockchain identity management system encompasses a wide array of technical, operational, regulatory, societal, and economic considerations. From a technical perspective, it involves the development of blockchain infrastructure, integration with existing systems, and implementation of cryptographic protocols. Operationally, it covers user onboarding, identity verification, and management processes. Regulatory compliance with data protection laws and industry standards is crucial, alongside addressing societal concerns about privacy and inclusivity. Economically, there are opportunities for streamlining processes and creating new business models. Security remains paramount throughout, requiring robust measures to protect against cyber threats and unauthorized access. In essence, the scope of a blockchain identity management system is broad and seeks to address various challenges while revolutionizing identity verification processes.

1.4 APPLICATIONS OF BLOCKCHAIN IDMS

Digital Identity Verification: Blockchain identity management systems can streamline the process of verifying digital identities for individuals, businesses,

and organizations. It enables secure and tamper-proof verification of personal information, reducing the risk of identity theft and fraud.

KYC/AML Compliance: In industries such as banking, finance, and cryptocurrency, Know Your Customer (KYC) and Anti-Money Laundering (AML) regulations require thorough identity verification processes. Blockchain identity management systems provide a robust and auditable solution for KYC/AML compliance, ensuring that users' identities are verified securely and efficiently.

Supply Chain Management: Blockchain-based identity management can enhance supply chain transparency and traceability by securely recording the identity and provenance of goods and components. It enables stakeholders to verify the authenticity and integrity of products throughout the supply chain, reducing counterfeiting and ensuring compliance with quality standards.

Decentralized Finance (DeFi): DeFi platforms leverage blockchain identity management systems to facilitate peer-to-peer transactions and financial services without the need for intermediaries. Users can securely manage their digital assets and access financial services such as lending, borrowing, and trading while maintaining ownership and control of their funds.

Academic Credential Verification: Blockchain-based systems securely issue, verify, and store academic credentials, ensuring the authenticity of academic achievements and facilitating credential verification.

CHAPTER 2

LITERATURE SURVEY

Several studies on Blockchain Identity Management System have been conducted in the recent years. This section summarizes all of the previous work on facial expression recognition.

2.1 INFERENCES FROM LITREATURE SURVEY

[1] A survey on blockchain-based identity management and decentralized privacy for per-sonal data

This is an overview of Identity management approaches, introducing an architecture overview as well as the relevant actors in such a system and blockchain technology as solution for distributed user-centric identity. Finally, we discuss the existing solutions and point out the research gaps and elaborate challenges and trade-offs towards building a complete identity management system.

[2] Revocable and Offline-Verifiable Self-Sovereign Identities

In this paper, the Self-Sovereign Identity (SSI) model to support both revocation as well as offline-verification. Our concept introduces attestations of validity for a point in time, which are issued by the SSI network for credentials that have not been revoked, i.e., added by authorized entities to a revocation list.

[3] A Comparative Survey on Blockchain-based Self Sovereign Identity System

In this paper, we intend to make an exhaustive study on different blockchain based self-sovereign identity implementations (such as Sovrin, Uport, EverID, LifeID, Sora, Self-Key) along with its architectural components and discuss about use case of self-sovereign identity.

[4] A blockchain-based per-sonal data and identity management system

In this research work, we propose a conceptual design and high-level architecture for a Blockchain-based Personal Data and Identity Management System (BPDIMS),

a human-centric and GDPR-compliant personal data and identity management system based on the blockchain technology. We describe how BPDIMS's architecture utilizes blockchain technology to provide a high-level of security, trust and transparency. We discuss how BPDIM's human-centric approach with GDPR compliance shifts the control over personal data to the end users and empowers them better.

[5] Deployment of a Blockchain-Based Self-Sovereign Identity

The proposed solution achieves passport-level legally valid identity. This solution for making identities Self-Sovereign, builds on a generic provable claim model for which attestations of truth from third parties need to be collected. The claim model is then shown to be both blockchain structure and proof method agnostic. Four different implementations in support of these two claim model properties are shown to offer sub-second performance for claim creation and claim verification. Through the properties of Self-Sovereign Identity, legally valid status and acceptable performance, our solution is considered to be fit for adoption by the general public.

2.2 EXISTING SYSTEM:

Identity Management System (IDMS) refers to how users or individuals are identified and authorized to use organizational systems and services. Since traditional identity management and authentication systems rely heavily on a trusted central authority, they cannot mitigate the effects of single points of failure. As a decentralized and distributed public ledger in a peer-to-peer (P2P) network, Blockchain (BC) technology has garnered a considerable amount of attention in the field of IDMS in recent years. Through Self-Sovereign Identity (SSI), users can have full authority over their digital identity. Successful implementation of a BC-based IDMS can significantly increase the degree of privacy and security of a user's SSI. However, the integration of BC-based IDMS to provide a user with SSI is still an unorganized area of research in its early stages of development. This article presents an extensive literature review of state-of-the-art academic publications as well as commercial market offerings regarding the applicability of BC-based SSI solutions. It also provides a detailed preliminary regarding the building blocks of blockchain technology and a progressive roadmap of IDMS solutions. In order to

develop an effective BC-based IDMS solution that focuses on securing a user's SSI, this article outline five essential components of a BC-based IDMS: authentication, integrity, privacy, trust, and simplicity. Furthermore, we perform a security analysis that outlines several types of adversarial threats that can cause potential damage to the BC-based IDMS. We identify and discuss associated issues and challenges by analyzing several notable BC-based IDMS solutions in academic literature. We also highlight potential research gaps and provide future research scope.

Inference of The Existing System:

The IDMS which is also known as identity management system this means looking at how customers or people are studied and allowed to use the company's systems and services. Since ordinary identification management and Authorization structures depend intensely on a relied on important authority and overlook to moderate the outcomes. Each calculate comes up short. It may be a decentralized and distributed peer-to-peer (P2P) arrange. Block Chain (BC) era has pulled in enormous intrigued on this respect. IDMS in a long time. With the Senior Self-Identity (SSI), clients can get total get to rights. Their computerized way of recognizing who they are. Using a BC-based IDMS can improve the security of a user's SSI. But the combination of IMDS and conveying SSI to the user is still being researched and developed. The article too presents a total writing evaluation of modern logical guides. With respect to the pertinence of BC-based SSI arrangements given interior the mechanical commercial center. It's the indistinguishable Gives an in-profundity creation to the building pieces of innovation. A progressive guide for IDMS arrangements. An effective IDMS reply built from BC. This article depicts the five transcendent variables in SSI client care. BC Based IDMS: Confirmation, Judgment, Secrecy, Believe and Straightforwardness. It is prepared to be connected A security assessment that depicts diverse antagonistic threats that have the potential to reason harm. BC is based on IDMS. We look at, see and talk current inconveniences and challenges there are a few high-quality BC-based IDMS arrangements inside the logical writing. It too highlights the potential Inquire about holes and give space for fate investigate.

2.3 PROBLEMS WITH EXISTING IDENTITY MANAGEMENT SYSTEMS

A digital identity is the total information about an individual or organization that exists online. Data that forms a digital identity includes usernames, buying history, ID number, and search history. Almost all of our digital identities are connected through devices, services, and apps that have dominantly used centralized and federated identity systems.

DISADVANTAGES OF EXISTING SYSTEM

- **Complexity:** Blockchain technology is inherently complex, requiring specialized knowledge and expertise for development, implementation, and maintenance. This complexity can be a barrier to adoption for organizations and individuals unfamiliar with blockchain concepts.
- **Scalability:** Current blockchain networks often face scalability issues, including limited transaction throughput and high latency, which can hinder the performance of identity management systems, especially during periods of high demand.
- **Privacy Concerns:** While blockchain offers enhanced security through encryption and decentralization, it also presents challenges in terms of privacy. All transactions on a public blockchain are visible to anyone, potentially exposing sensitive identity information to unauthorized parties.
- **Regulatory Uncertainty:** The regulatory landscape surrounding blockchain technology and identity management is still evolving, leading to uncertainty and potential compliance challenges for organizations operating in regulated industries.
- **Centralization Risks:** While blockchain is designed to be decentralized, the infrastructure supporting blockchain networks, such as mining pools and node operators, can become centralized over time. This centralization introduces vulnerabilities and risks to the overall security and integrity of the system.

Addressing these disadvantages and challenges is crucial for the widespread adoption and success of blockchain identity management systems. Solutions may involve advancements in blockchain technology, regulatory frameworks, interoperability standards, and user interface design.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

3.2 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in

order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

3.3 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system

analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justify. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - Java
- IDE - Net beans
- Database - MYSQL

3.4 JAVA

Java is a versatile, platform-independent programming language renowned for its simplicity and robustness. With its "write once, run anywhere" capability, Java allows developers to create applications that can seamlessly run on any system with a Java Virtual Machine. Its object-oriented nature, automatic memory management, and extensive standard library make it a popular choice for building a wide range of software solutions, from desktop applications to enterprise systems and Android mobile apps. Additionally, Java prioritizes security, featuring built-in mechanisms to ensure code integrity and protect against vulnerabilities, further solidifying its position as a cornerstone of modern software development.

JAVA Features

- **Platform Independence:** Java's "write once, run anywhere" capability allows code to be executed on any platform with a Java Virtual Machine (JVM), ensuring portability across diverse environments.
- **Object-Oriented:** Java is a fully object-oriented language, supporting concepts like classes, objects, inheritance, polymorphism, encapsulation, and abstraction, facilitating modular and reusable code.

- **Automatic Memory Management:** Java features automatic garbage collection, which manages memory allocation and deallocation, helping to prevent memory leaks and buffer overflows.
- **Rich Standard Library:** Java provides a comprehensive standard library (Java API) that includes classes and interfaces for performing various tasks such as input/output operations, networking, data manipulation, concurrency, and more, enabling developers to build applications efficiently.
- **Concurrency Support:** Java supports multi-threading and concurrency through features like threads, synchronization, and concurrent utilities such as the Executor Framework and Concurrent Collections, enabling the development of scalable and responsive applications.
- **Exception Handling:** Java has a robust exception handling mechanism, allowing developers to handle errors and exceptional conditions gracefully using try-catch blocks, throw statements, and exception classes.
- **Security Features:** Java prioritizes security with built-in mechanisms such as bytecode verification, a security manager, and cryptographic APIs, ensuring code integrity and protecting against vulnerabilities like buffer overflows and unauthorized access.
- **Dynamic Binding and Reflection:** Java supports dynamic binding and reflection, allowing objects to be bound to methods at runtime and enabling introspection and manipulation of class objects, providing flexibility and extensibility.
- **Networking Capabilities:** Java provides robust networking capabilities through classes like Socket and ServerSocket, enabling developers to create client-server applications, communicate over networks, and implement protocols such as TCP/IP and UDP.
- **GUI Development:** Java offers tools and libraries like Swing and JavaFX for building graphical user interfaces (GUIs), allowing developers to create rich and interactive desktop applications with ease.

3.5 SQL (Structured Query Language)

SQL (Structured Query Language) is a powerful tool for managing relational databases, offering commands for data manipulation, querying, and database

administration. It allows users to create and modify database structures, execute complex queries to retrieve specific data, and enforce data integrity through constraints and transactions. Widely used in various industries, SQL is essential for database administrators, developers, and data analysts for efficiently working with relational databases.

SQL Features

Database Creation: SQL allows users to create and manage relational databases, including defining the structure of tables, specifying data types, and establishing relationships between tables.

Data Manipulation: SQL provides commands for inserting, updating, deleting, and retrieving data from a database. These commands include INSERT, UPDATE, DELETE, and SELECT, which enable users to perform various data manipulation operations.

Querying: SQL is primarily used for querying databases to extract specific information. The SELECT statement is used to retrieve data from one or more tables based on specified criteria, allowing users to filter, sort, and aggregate data as needed.

Data Definition Language (DDL): SQL includes commands for defining and modifying the structure of database objects such as tables, indexes, views, and constraints. DDL commands include CREATE, ALTER, and DROP, among others.

Data Control Language (DCL): SQL provides commands for controlling access to the database and managing user permissions. DCL commands include GRANT and REVOKE, allowing administrators to grant or revoke privileges to users and roles.

Data Integrity: SQL supports data integrity constraints such as primary keys, foreign keys, unique constraints, and check constraints, which help ensure the accuracy and consistency of data within the database.

Transactions: SQL supports transactions, which are units of work that are executed and committed or rolled back as a single, atomic operation. Transaction control commands such as COMMIT, ROLLBACK, and SAVEPOINT are used to manage transactions.

3.6 HTML

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create and design web pages. It consists of a set of elements and tags that define the structure and content of a web document.

HTML Features

Element Structure: HTML provides a structured way to organize content on a web page using various elements such as headings, paragraphs, lists, links, images, tables, and forms. This element structure defines the layout and presentation of the web page.

Hyperlinks: HTML allows the creation of hyperlinks using the <a> (anchor) element, enabling users to navigate between different web pages or sections within the same page. Hyperlinks are essential for building interconnected web content and facilitating user interaction.

Forms: HTML supports the creation of interactive input forms using elements like <form>, <input>, <select>, <textarea>, and <button>. Forms enable users to submit data to a web server for processing, facilitating tasks such as user authentication, data submission, and user feedback.

Semantic Markup: HTML includes semantic elements that provide meaning and structure to the content they enclose. Semantic elements like <header>, <footer>, <nav>, <article>, <section>, and <aside> help browsers, search engines, and assistive technologies understand the purpose and context of the content, improving accessibility and search engine optimization (SEO).

3.7 CSS (CASCADING STYLE SHEET)

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS Features

Selectors and Declarations: CSS allows developers to select HTML elements based on their attributes, classes, IDs, or relationships with other elements using selectors. Once selected, developers can apply style declarations to these elements, specifying properties like color, font, size, margin, padding, and more.

Cascading Style Rules: CSS employs a cascading mechanism to determine the final style applied to an element. This mechanism allows styles to cascade from parent elements to their children, with styles defined closer to the target element taking precedence over more general styles. Additionally, CSS supports inheritance, where child elements inherit styles from their parent elements unless explicitly overridden.

Layout and Positioning: CSS provides powerful tools for controlling the layout and positioning of elements within a web page. Developers can use properties like display, position, float, flexbox, and grid to create complex layouts, align elements, and control their placement on the page.

Responsive Design: CSS enables developers to create responsive web designs that adapt to different screen sizes and devices. Media queries allow developers to apply different styles based on factors like screen width, height, resolution, and orientation, ensuring that web pages display optimally on various devices, from desktops and laptops to tablets and smartphones.

3.8 JAVASCRIPT

JavaScript is a versatile scripting language widely used for client-side web development. Its ability to manipulate the Document Object Model (DOM) allows for dynamic content updates and interactive features on web pages. JavaScript's event-

driven paradigm, DOM manipulation capabilities, and support for asynchronous programming make it indispensable for creating dynamic and engaging web applications.

JavaScript Features

Event-Driven Programming: JavaScript follows an event-driven programming paradigm, where actions or events trigger the execution of code. Developers can define event handlers to respond to user interactions like clicks, key presses, mouse movements, and form submissions, enabling dynamic and interactive web experiences.

DOM Manipulation: JavaScript enables manipulation of the Document Object Model (DOM), which represents the structure of a web page as a hierarchical tree of objects. Developers can use JavaScript to access, modify, or create DOM elements dynamically, allowing for dynamic content updates, animations, and interactive features on web pages.

Asynchronous Programming: JavaScript supports asynchronous programming, allowing code to execute non-blocking operations such as fetching data from servers, performing calculations, or executing I/O operations without halting the execution of the main program. Asynchronous features like Promises, `async/await`, and callback functions enable developers to handle asynchronous tasks efficiently and manage complex workflows in web applications.

3.9 BOOTSTRAP

Bootstrap is a popular front-end framework for building responsive and mobile-first websites and web applications. Developed by Twitter, Bootstrap provides a comprehensive set of pre-designed CSS styles, JavaScript plugins, and responsive layout components. It simplifies web development by offering a grid system, typography, forms, buttons, navigation bars, and other UI components that can be easily customized and integrated into projects. With its extensive documentation and wide range of features, Bootstrap empowers developers to create modern, visually appealing, and cross-browser compatible websites efficiently.

Ajax:

Ajax, short for Asynchronous JavaScript and XML, is a web development technique that enables asynchronous communication between a web browser and a server. It allows for updating parts of a web page without needing to reload the entire page, enhancing user experience and responsiveness in web applications.

3.10 LIBRARIES AND CLASSES USED

MailcapCommandMap: `MailcapCommandMap` is a class in Java that provides a mapping between MIME types and commands for handling data of those types. It is part of the JavaMail API, which is used for sending, receiving, and manipulating email messages. `MailcapCommandMap` allows developers to specify which applications or handlers should be invoked for different MIME types, such as opening a PDF file with a PDF viewer or playing an audio file with a media player. This class provides methods for adding or retrieving command mappings and is commonly used in email client applications to customize how various MIME types are handled.

ZXing (Zebra Crossing): The library provides functionalities to decode barcodes from images captured by a camera or stored as image files. It also offers utilities for generating barcode images from text or data. Developers can integrate the ZXing library into their Java applications to perform barcode-related tasks, such as scanning product barcodes, processing tickets or coupons, or generating QR codes for sharing information.

Web Servlet: A web servlet in Java is a server-side component used to handle HTTP requests and generate dynamic responses in web applications. It follows a lifecycle model comprising initialization, service, and destruction phases, with developers able to override methods to customize behavior. Servlets process various types of HTTP requests, extracting data from parameters and headers to execute business logic and generate dynamic content, making them fundamental for building interactive web applications in Java.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR MODEL

The process of identity management ensures that only authorized users or entities have access to certain resources or systems by verifying their identities. Data security is essential in ensuring that only authorized users can access confidential information. Blockchain is a system for managing records that lacks control from one single entity. It keeps track of transactions on a safe and open network. The way blockchain works makes it hard for hackers to take over because no one person or group has all the control over the information. Blockchain-based identity systems help users have more control over their personal information, and they can decide who gets to see it. They can make sure their information is kept safe and cannot be changed. Blockchain-based identity management not only makes data safer, but it also brings other advantages. For instance, it could make it easier for online services to check your identity, so you don't have to keep giving your personal information to different companies. It could also help people keep their information more private because they could decide how their data is shared and used.

Blockchain technology can help prevent people from stealing your identity in many ways:

- Decentralized Identity Verification
- Self-Sovereign Identity
- Immutable Records
- Fraud Detection

ADVANTAGES OF PROPOSED SYSTEM

- **Enhanced Security:** Blockchain identity management systems leverage cryptographic techniques to secure identity data, reducing the risk of data

breaches and unauthorized access. The decentralized nature of blockchain adds an extra layer of security by eliminating single points of failure.

- **Decentralization:** By decentralizing identity data storage and verification processes, blockchain identity management systems reduce reliance on centralized authorities, minimizing the risk of manipulation or tampering with identity records.
- **Privacy Control:** Users have greater control over their personal identity data in blockchain-based systems, allowing them to selectively share information and maintain privacy. This helps mitigate privacy concerns associated with centralized identity management systems.
- **Efficiency:** Blockchain streamlines identity verification processes by providing a single source of truth for identity records. This reduces duplication of effort and administrative overhead, leading to more efficient identity management workflows.
- **Immutability:** The immutable nature of blockchain ensures the integrity and tamper-resistance of identity records. Once recorded on the blockchain, identity information cannot be altered or deleted without consensus from the network participants, enhancing trust and reliability.
- **Transparency:** Blockchain's transparent and auditable nature fosters trust among users and stakeholders in the identity management ecosystem. All transactions and changes to identity records are visible to authorized parties, promoting accountability and integrity.
- **Interoperability:** Blockchain identity management systems have the potential to enable interoperability between different organizations, platforms, and systems. This facilitates seamless identity verification and authentication processes across various applications and domains.
- **Reduced Identity Theft:** The use of cryptographic signatures and decentralized storage mechanisms in blockchain identity management systems helps mitigate the risk of identity theft and fraud, as identity data is securely stored and controlled by individual users.

4.2 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED SYSTEM

This system is designed by integrating Java, SQL, HTML, CSS, JavaScript, Blockchain Algorithms and Bootstrap into the Blockchain Identity Management System, these are used to build a comprehensive solution that combines backend processing capabilities with a user-friendly frontend interface. This approach enables efficient identity management, seamless user interaction, and a consistent user experience across different platforms and devices.

The testing plan for this Blockchain Identity Management system can include several stages, such as:

- **Unit testing:** This involves testing the individual components of the system, such as the data collection and pre-processing, model development and training, and model evaluation and validation, to ensure that they are working as expected.
- **Integration testing:** This involves testing the integration of the different components of the system to ensure that they are working together as expected.
- **Performance testing:** This involves testing the performance of the system, such as the accuracy, precision, and recall of the model, to ensure that it meets the required performance metrics.
- **Acceptance testing:** This involves testing the system in a real-world setting with end-users to ensure that it is meeting their needs and requirements.

Overall, the software for implementation and testing plan of the proposed Blockchain Identification Control: Sparking A Data Protection Revolution system using AES and SHA algorithms should be designed to ensure that the system is accurate, efficient, and reliable, and can be deployed and maintained effectively.

4.3 SYSTEM ARCHITECTURE:

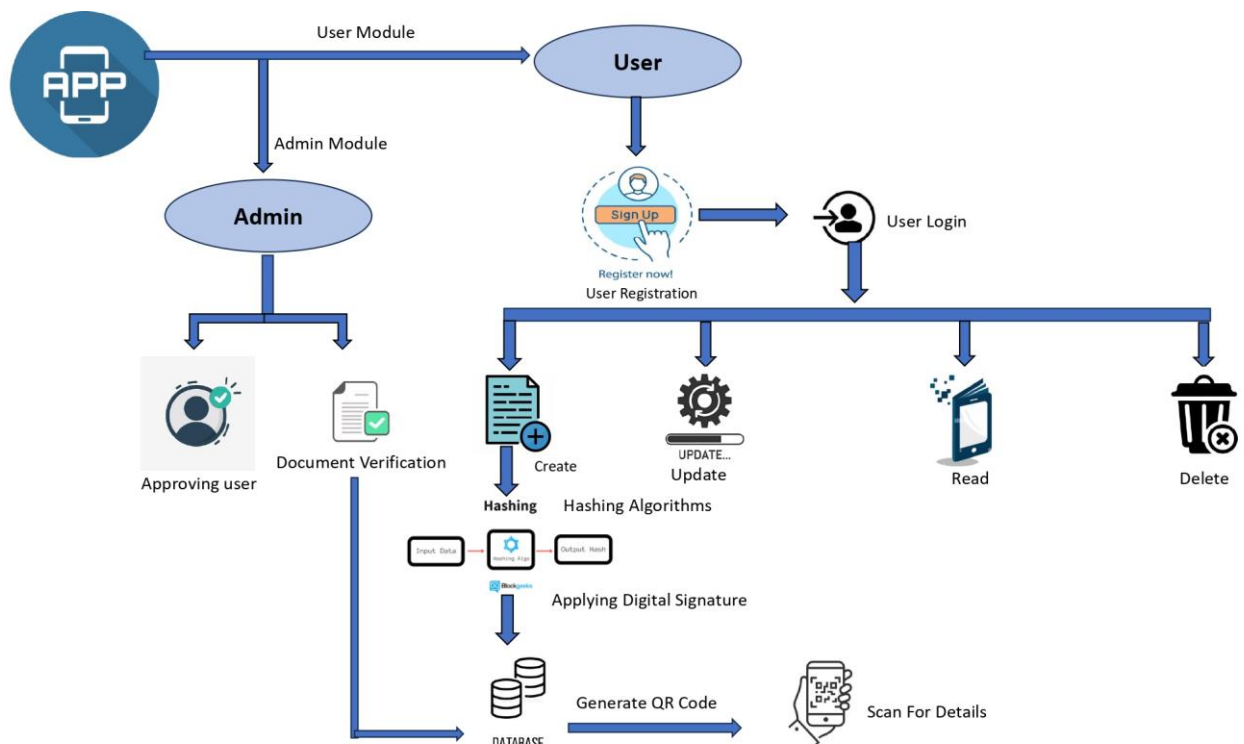


Fig 4.1 – Architecture Diagram

In this system, the user can verify the user's activity and verify the accounts. The admin has access to all of the user's documents that have been uploaded to the system. To log in to the system, the user must first register; once the account has been verified by the admin, the user can log in with a username and password. The user can update the profile as needed. The user can view if logs have been updated or deleted, as well as the status of fabricated or altered document details.

4.4 WORKFLOW FOR BLOCKCHAIN IDMS

User Registration:

Users initiate the registration process by providing identity information such as name, email, and biometric data. The system verifies the provided information and assigns a unique identifier to the user. Identity data is encrypted and stored on the blockchain along with a digital signature to ensure integrity.

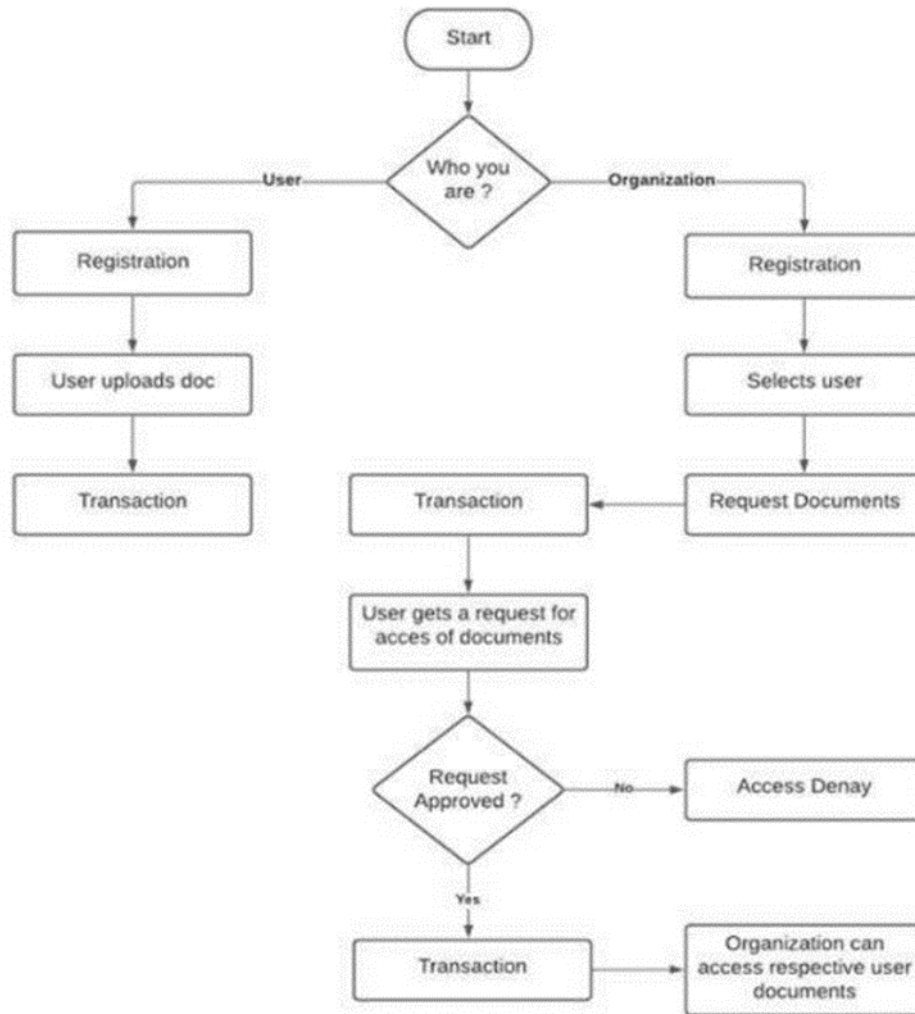


Fig 4.2: - Work flow Diagram

Identity Authentication:

When accessing services or applications, users authenticate themselves using their unique identifier and authentication methods such as passwords, biometrics, or cryptographic keys. The system verifies the user's identity by retrieving relevant identity data from the blockchain and comparing it with the authentication credentials provided.

Identity Updates:

Users can update their identity information or add new attributes over time. Updated information undergoes verification and validation processes before being recorded on the blockchain to ensure accuracy and integrity.

Access Control:

Service providers or applications query the blockchain to verify the identity of users before granting access to resources or services. Access control policies can be enforced based on the user's verified attributes and permissions stored on the blockchain.

Audit Trail:

All transactions related to identity management, including registrations, verifications, updates, and access requests, are recorded as immutable entries on the blockchain. Audit logs provide a transparent and tamper-proof record of all activities, ensuring accountability and compliance with regulations.

Revocation and Deletion:

In case of compromised credentials or user requests for deletion, revocation mechanisms are employed to invalidate access privileges and remove sensitive information from the blockchain. Revocation certificates or deletion records are appended to the blockchain to ensure transparency and compliance with data protection regulations.

Governance and Consensus:

Governance mechanisms are established to manage the operation and evolution of the blockchain identity management system. Consensus algorithms ensure agreement among network participants on the validity of identity transactions and updates recorded on the blockchain.

Continuous Improvement:

The system undergoes periodic audits, reviews, and updates to address security vulnerabilities, enhance functionality, and adapt to evolving regulatory requirements and industry standards.

By following this workflow, a Blockchain Identity Management System ensures secure, transparent, and interoperable identity management processes while maintaining user privacy and data integrity.

4.5 MODULES:

This Blockchain IDMS has four proposed modules:

- Admin Module
- User Module
- CRUD Module (create, read, update, delete)
- Documentation Module.

ADMIN MODULE:

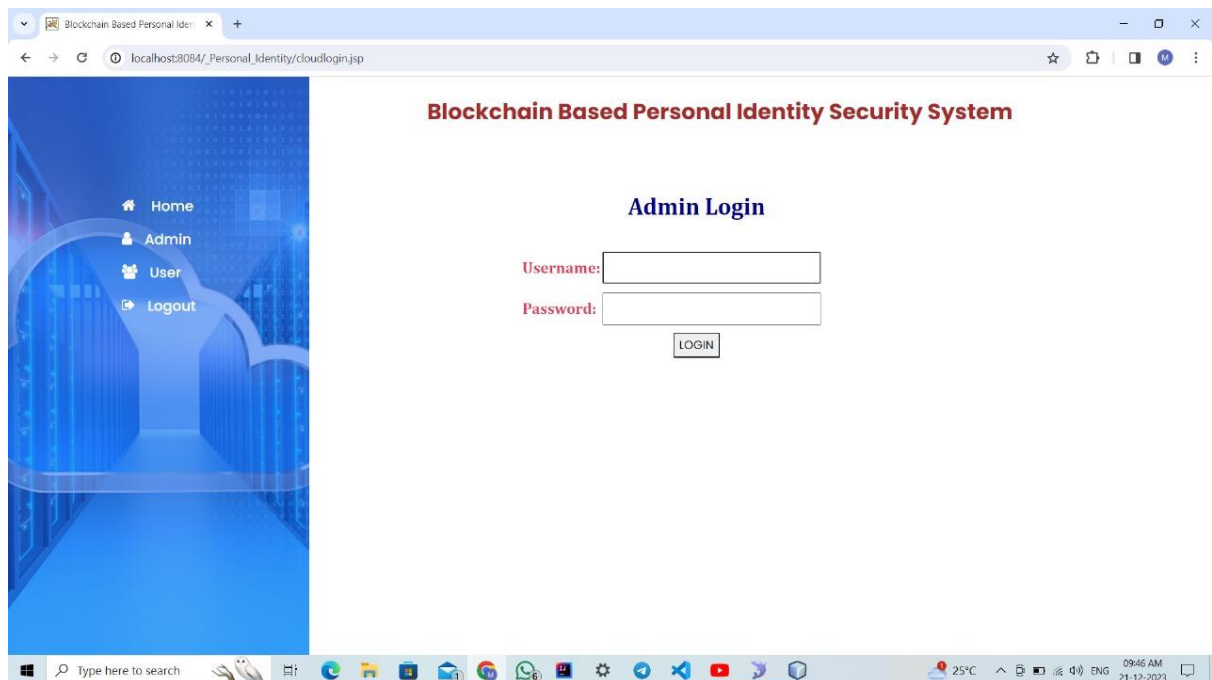


Fig 4.3: – Admin Login Page

ADMIN MODULE:

The admin module envelops a basic feature of web applications, highlighting a secure login page as its portal. Confirmation measures, counting strong watchword

hashing and shields against brute-force assaults, guarantee that as it were authorized directors pick up get to. Upon fruitful login, chairmen are welcomed with a comprehensive dashboard, advertising experiences into system activities and pending errands. One pivotal function is client confirmation, enabling admins to survey and endorse client enlistments, guaranteeing compliance with stage guidelines. Furthermore, the module prepares directors to scrutinize records and records transferred by clients, evaluating adherence to arrangements and confirming genuineness. Client administration devices, counting account creation, upgrades, and deactivation, allow control over client get to. A review path records admin exercises, cultivating responsibility, whereas a notice framework keeps chairmen educated of pivotal occasions such as modern enrolments or pending archive surveys. Security contemplations, versatility, and ideal execution are foremost within the module's plan, with intermittent upgrades and reviews fundamental for keeping up framework judgment. The module's capacity to encourage secure communication with clients improves its viability in overseeing and confirming user-generated substance. The admin module plays a essential part in keeping up the integrity and security of the whole application. The client confirmation handle frequently includes a fastidious audit of user-submitted documentation and may include communication with clients to address any errors or extra verification necessities. This coordinate interaction cultivates transparency and trust between administrators and clients.

- ***Login***

The admin can log in using a username and password.

- ***View Users***

The account of the user is verified by the admin.

The admin has access to all of the documents that the user has uploaded.

They have access to all of the user's logs.

The admin can view the status of the fabrication.

All the complaints can be viewed by the admin.

USER MODULE:

The user module may be an essential component of web applications that rotates around client enlistment, verification, and document/file administration. At its core,

the module starts with a user-friendly registration process where people give fundamental data to make an account. This regularly includes entering a one-of-a-kind username, a secure secret word, and may incorporate extra subtle elements for profile completion. Upon effective enrolment, clients pick up get to a secure login page, where verification instruments guarantee the assurance of their account. Security measures such as watchword hashing and account lockout approaches contribute to a strong verification framework, shielding client qualifications. Once logged in, users are enabled to transfer records and reports to the stage. An instinctive and responsive client interface encourages the transfer prepare, permitting clients to yield different record designs whereas following to estimate impediments and platform-specific rules. To improve the user involvement, real-time criticism amid the transfer prepares guarantees straightforwardness and decreases the probability of mistakes. The module gives users with the capability to see and oversee their transferred records or archives. This incorporates highlights like record organization, form control, and the capacity to upgrade or evacuate records as required. Also, users may have get to a point by point history of their record intelligent, advertising bits of knowledge into transfer dates, adjustments, or any activities performed on their reports. in conclusion, the user module could be a comprehensive arrangement that consistently coordinating client enrolment, verification, and record administration functionalities. Its user-centric plan, coupled with strong security measures, enables people to associated with the stage proficiently, transfer and oversee their records, and contribute to a collaborative and secure computerized environment. Regular updates and user criticism components contribute to the ceaseless improvement of the client module's capabilities.

- ***Register***

The user needs to register with their basic details.

- ***Login***

After account verification, the user can log in to the system.

- ***Profile***

The user can add or update their profile details.

- ***Change Password***

They can also change their password to the new one.

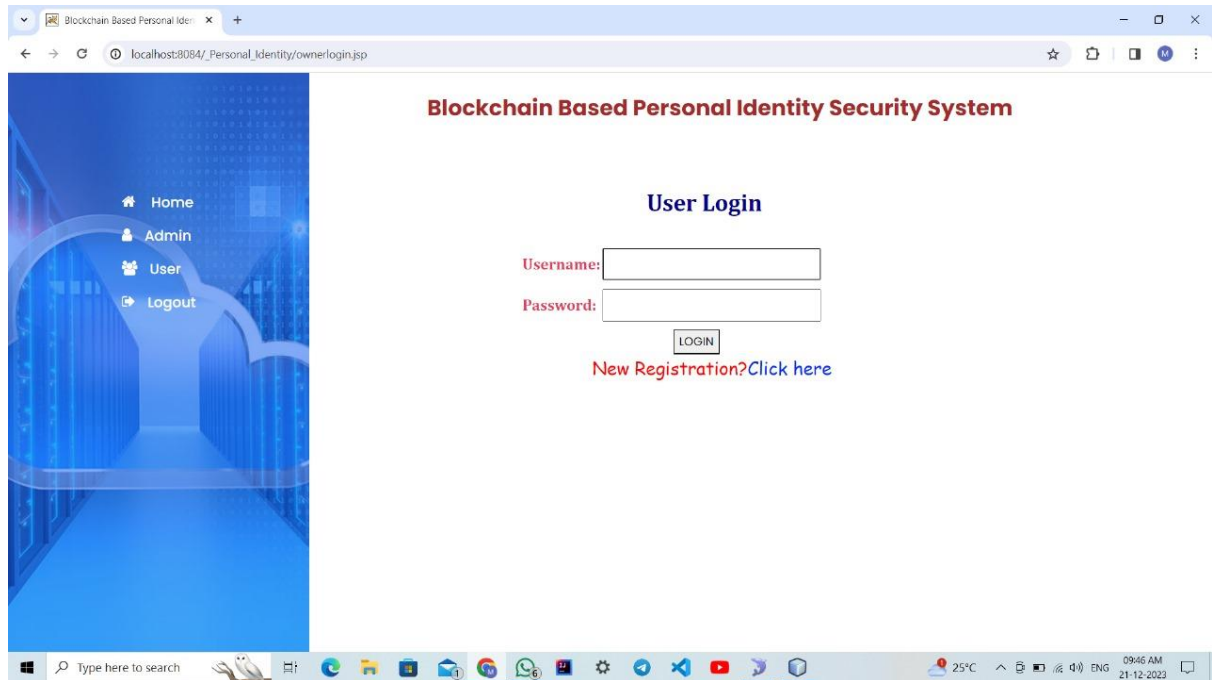


Fig 4.4: – User Login Page

Manage Documents

- The user can add, update or delete documents in the system.
- The documents the user uploads are verified by the admin.
- The user can view the QR Code.
- They can also check the status of documents that are being fabricated.

View logs

- The user can see if logs have been updated or deleted, as well as the status of document details being fabricated.
- If the user discovers a fabrication or an action that was not performed by them, they can notify the admin.

Scan QR Code

- The user can scan the QR Code and view all the details.

Verify Document

- They can add all the details & verify with the ledger.

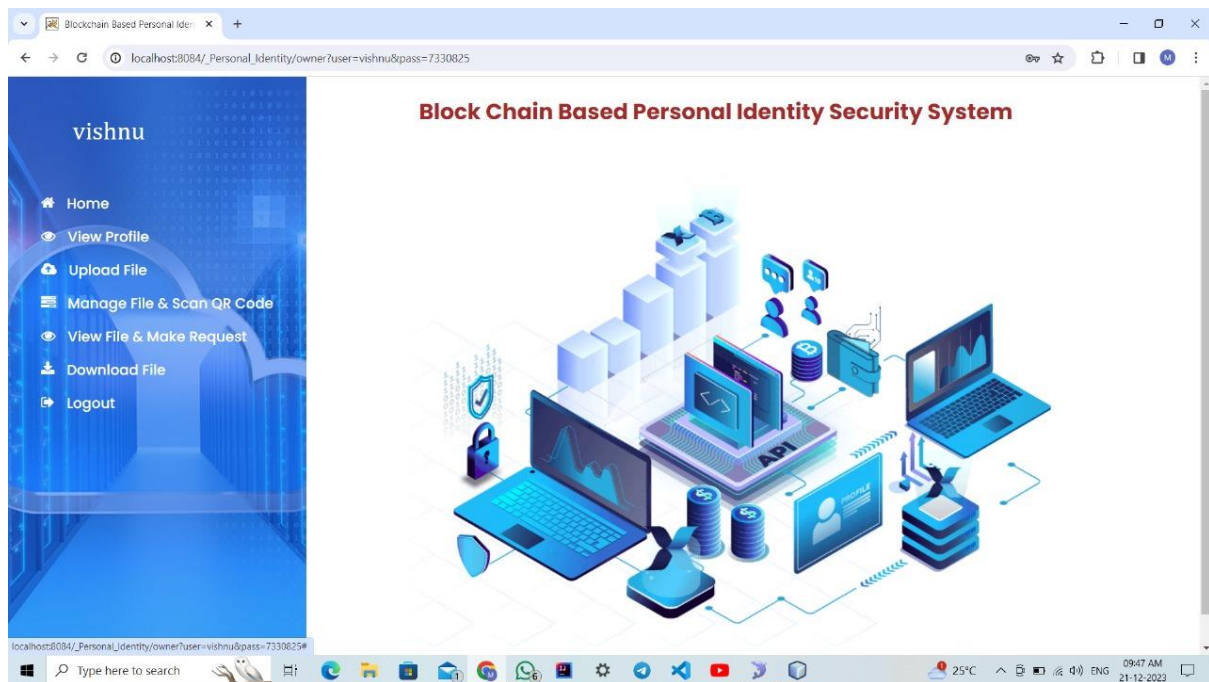


Fig 4.5: – User Module

CRUD MODULE:

The crud (create, read, update, erase) module is a basic component inside web applications, offering clients the capability to associated comprehensively with information taking after confirmation. Upon logging in, clients can consistently make unused records by contributing significant data through user-friendly shapes. The studied operation empowers clients to see both their made substance and existing records, giving a look and channel usefulness for effective information recovery. With the upgrade highlight, clients can alter existing information, guaranteeing that as it were authorized people can make changes and joining approval checks to preserve information exactness. The erase operation engages clients to evacuate obsolete or undesirable records, including affirmation instruments to avoid inadvertent erasures. The module's client interface is planned for ideal client encounter, joining clear plan components and instructive input messages. A review path records client activity, improving straightforwardness and responsibility. Exacting security measures, counting verification, authorization, and information approval, defend against unauthorized get to or pernicious exercises. Execution optimizations guarantee versatility, making the crud module a flexible and crucial apparatus for secure and productive information administration inside web applications.

4.6 ALGORITHMS AND MECHANISMS USED

POA Consensus Mechanism:

In the ProofOfAuthority mechanism, a limited number of individuals are responsible for validating transactions and keeping the blockchain up to date. It works like this: according to the chosen plan, one or more approving machines can create each new block of transactions that will be added to the Blockchain. Depending on the configuration of the Blockchain, a new block can be added without validation if it is approved by most of the generators or users.

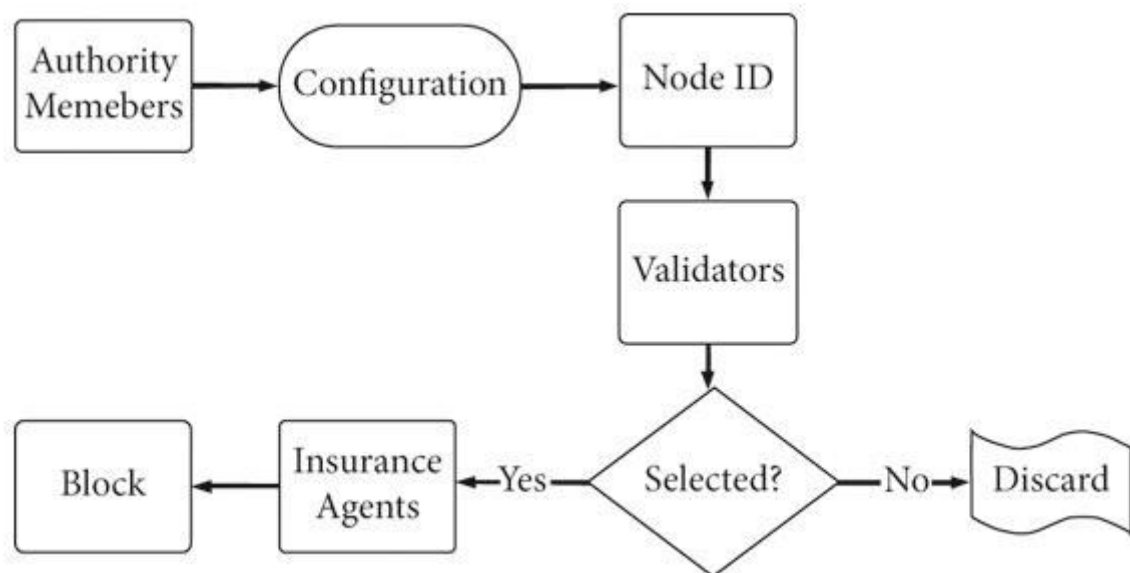


Fig 4.6: - POA Consensus Mechanism

SHA-256(SECURE HASH ALGORITHM):

The SHA 256 algorithm could be an unmistakable cryptographic hash function in advanced security. Its strength and resistance to collision attacks make it a favored choice for different applications, extending from advanced signatures and blockchain innovation to password hashing and information integrity verification. SHA-256 produces a fixed-size hash esteem of 256 bits, guaranteeing a tall degree of uniqueness and irreversibility, hence encouraging secure

information transmission and capacity. The algorithm widespread appropriation in different businesses underscores its noteworthiness in defending delicate data and bracing cybersecurity measures. As innovation progresses, the significance and application of SHA 256 stay instrumental in maintaining the principles of confidentiality, integrity, and authenticity within the digital landscape. SHA-256 works as a one-way hash work, guaranteeing that turning around the hash to get the initial information is computationally unfeasible.

SHA-256 cannot be broken and proceeds to be widely used for cryptographic purposes. In any case, indeed in spite of the fact that no prompt vulnerabilities are known for SHA-256, the potential for future improvements in computing seem affect its security. Consequently, remaining upgraded with the most recent improvements in cryptographic investigate is advisable.

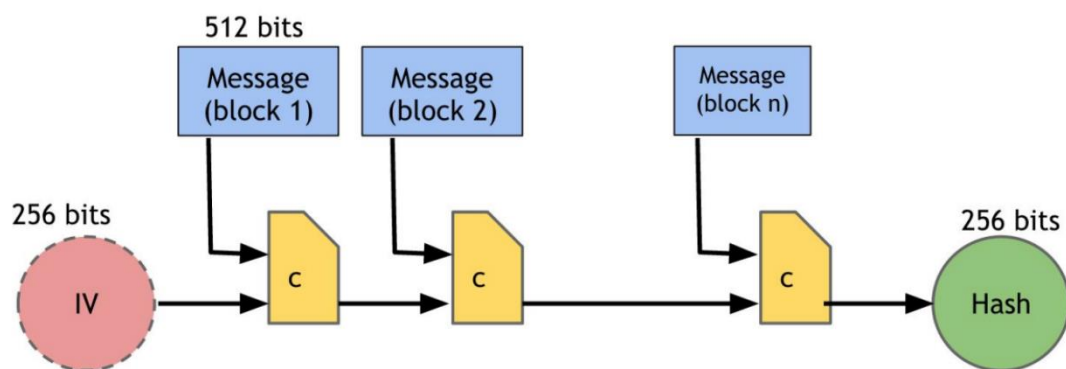


Fig 4.7: -SHA 256 Block Diagram

AES (Advanced Encryption Standard):

The Advanced Encryption Standard (AES) is a type of method to keep information safe. It is very strong and doesn't take too much time to use. Data in AES is typically processed in chunks that are a set size, usually 128 bits, and the keys used are also 128, 192, or 256 bits long. The network utilized in the process is an SPN, consisting of numerous rounds of substitution and permutation. The number of repetitions in the process is determined by the size of the key: 10 for a 128-bit key, 12 for a 192-bit key, and 14 for a 256-bit key. AES encryption has many parts. First, a key is expanded. Then there are some initial operations in the first

round. After that, there are many rounds of substitution, permutation, and mixing operations. Finally, there is a last round. Each stage of the process uses different math changes like SubBytes, ShiftRows, MixColumns, and AddRoundKey. These actions make things difficult to understand and spread them out, which are important for making encryption strong. AES decryption is like undoing the encryption. It uses the opposite steps in the opposite order.

AES Algorithm Working

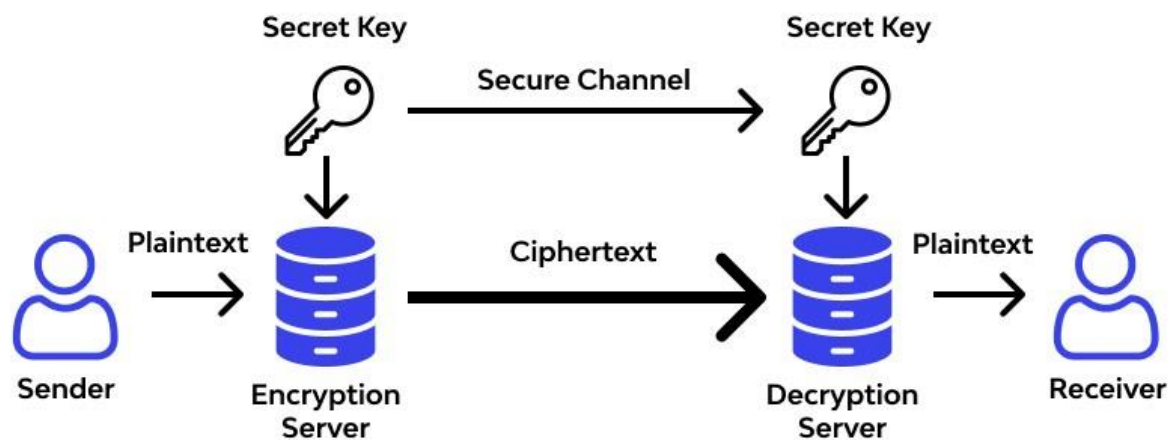


Fig 4.8: - AES Algorithm Working

Cipher Helper:

Cipher Helper class that facilitates data encryption and decryption utilizing the DES (Data Encryption Standard) algorithm. The class incorporates methods for encryption (cipher) and decryption. The secret key for these operations is set inside the class. The code also features helper methods to change over between byte arrays and strings. Within the primary strategy, a secret key and information are initialized, but the encryption and decoding methods are not conjured. There's a minor error within the primary method, where the factors secret Key and information are pronounced but not utilized. It is suggested to expel the redundant task of secret Key inside the course and to upgrade the most strategy by conjuring the encryption and decoding strategies for a viable exhibit. Also, the utilize of DES, whereas showcased within the code, is considered unreliable for numerous applications, and a more secure algorithm like AES is recommended for cryptographic purposes.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 IMPLEMENTATION

The homepage serves as the central center for getting to the blockchain personality administration framework, giving consistent route and get to fundamental functionalities. At its center, the homepage unmistakably highlights choices for client login, admin login, logout, and a domestic button, guaranteeing instinctive client interaction. For clients, the login alternative offers a secure door to get to personalized administrations and oversee their personalities inside the framework. Through vigorous verification instruments, clients can safely confirm themselves, picking up get to highlights like personality creation, confirmation, and administration. At the same time, the logout alternative permits clients to safely end their sessions, improving protection and information security. So also, admin is given with a committed login alternative, giving them advantaged get to regulatory functionalities and framework controls. Admin login guarantees legitimate oversight and administration of client characters, authorizations, and framework arrangements. Chairmen can use these capabilities to screen framework movement, resolve issues, and uphold security arrangements successfully. The nearness of a domestic button offers clients and chairmen alike a helpful implies to explore back to the homepage from any area of the framework. This determined route component upgrades convenience and guarantees a consistent client involvement, empowering quick moves between distinctive framework functionalities and screens. This project is mainly based on the Blockchain technology and this project is built on the top of the blockchain. This system will be used by the admin and as well as users.

The [Fig-B.1] is the home page of the project where the admin as well as user are able to login to the system by entering their credentials which they created earlier for the purpose of login.

User or client after entering their credentials can enter into the system [Fig-B.5] and they can able perform certain operations such as viewing their profile, uploading the file, managing the file, scanning the QR code for getting the uploaded details, able to view the uploaded data or records and download the files or records uploaded.

Admin after entering their credentials can enter into the admin login page [Fig-B.3] where he can able to do some operations like viewing the user activity, able to manage the user's requests, count of the uploaded content, able to view the files, attribute details, graph or performance of the system and finally he can able to logout from the system.

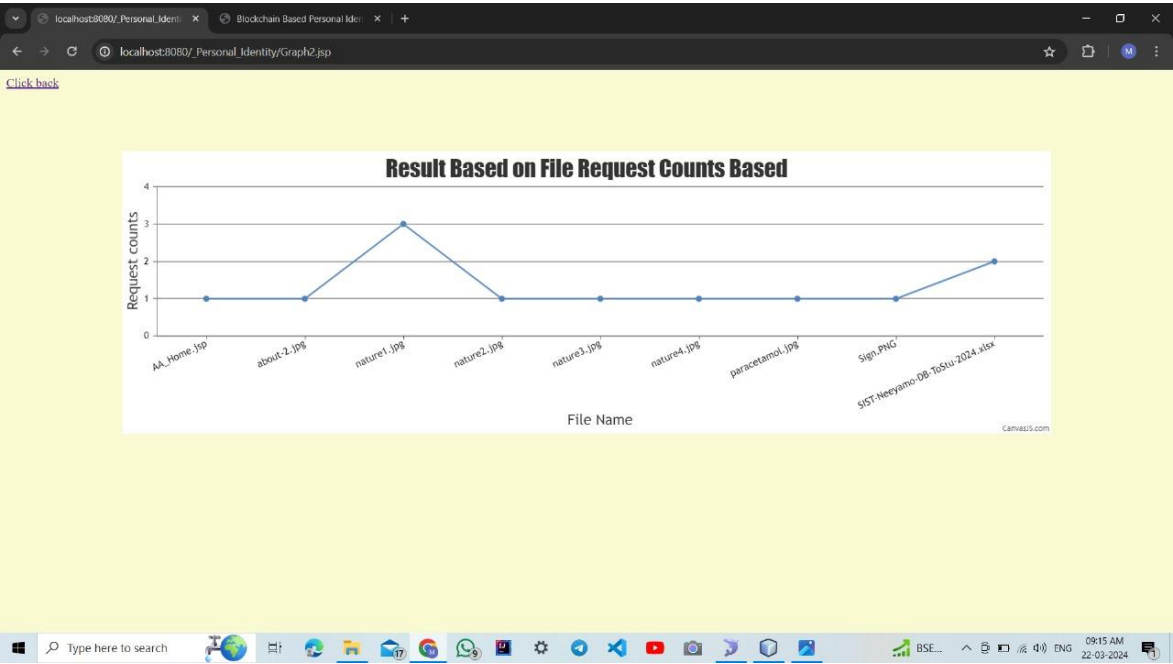


Fig 4.9: - File Request Count

The graph [Fig: 4.9] depicts the file request count over a specified period, offering insights into the demand for accessing files within a system. By visualizing this data, administrators can discern trends, identify peak usage times, and allocate resources accordingly. This graph aids in optimizing file management processes, ensuring efficient handling of user requests and maintaining system performance. Moreover, it facilitates proactive decision-making, allowing administrators to anticipate spikes in demand and implement preemptive measures to mitigate potential bottlenecks. Ultimately, the graph serves as a valuable tool for monitoring and managing file access within the system, enhancing overall efficiency and user satisfaction.

CHAPTER 6

RESULT AND DISCUSSION

The implementation of the Blockchain Identity Management System showcases compelling outcomes, notably in bolstering security, decentralization, and user privacy. Leveraging cryptographic techniques ensures data integrity and confidentiality, mitigating risks of unauthorized access. Decentralization reduces reliance on centralized authorities, fostering resilience and trust. Users gain control over their identity information, enhancing privacy and selective sharing capabilities. However, challenges like scalability and regulatory compliance necessitate ongoing research and refinement. Overall, the system presents a robust foundation for secure and user-centric digital identity solutions, poised to address evolving challenges in the digital landscape.

However, challenges such as scalability, interoperability, and regulatory compliance remain areas of concern. Scalability issues may arise as the system scales to accommodate a larger user base and increased transaction volume. Interoperability challenges may hinder the seamless integration of the system with legacy identity management systems and third-party applications. Additionally, ensuring compliance with evolving regulatory frameworks presents a complex and ongoing task for system developers and administrators.

Overall, the Blockchain Identity Management System represents a significant advancement in identity management practices, offering a secure, decentralized, and user-centric approach to identity verification and authentication. Continued research, collaboration, and innovation are essential to address remaining challenges and unlock the full potential of the system in revolutionizing digital identity ecosystems. As the system matures and evolves, it holds the promise of reshaping the way identities are managed and verified in the digital age, paving the way for a more secure, efficient, and user-friendly identity management landscape.

CHAPTER 7

CONCLUSION

In conclusion, the Blockchain Identity Management System represents a significant step forward in the realm of digital identity management. By harnessing the power of blockchain technology, the system offers a comprehensive solution that addresses critical challenges such as security, privacy, and decentralization. Through the use of cryptographic techniques and decentralized storage, it ensures the integrity and confidentiality of user identity data, reducing the risk of data breaches and unauthorized access. Moreover, the system empowers users with greater control over their identity information, enabling selective sharing and enhancing privacy protection.

Furthermore, the decentralized nature of blockchain mitigates reliance on centralized authorities, promoting resilience and trust in identity management processes. However, while the system demonstrates considerable potential, challenges such as scalability, interoperability, and regulatory compliance remain areas of concern. Addressing these challenges will require ongoing research, collaboration, and innovation within the field.

Nevertheless, the Blockchain Identity Management System lays a solid foundation for secure and user-centric digital identity solutions. Its implementation opens up new possibilities for enhancing identity management practices across various industries, including finance, healthcare, and government services. Moving forward, continued exploration and refinement of the system will be essential to realize its full potential and address emerging challenges in the digital age. Overall, the system represents a significant advancement in identity management technologies, poised to reshape the way identities are managed and verified in the modern digital landscape.

REFERENCES

- [1] Kuperberg M, "Blockchain-based identity management: A survey from the enterprise and ecosystem perspective," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1008–1027, Nov. 2020.
- [2] Dunphy P and Petitcolas F, "A first look at identity management schemes on the blockchain," *IEEE Security Privacy*, vol. 16, no. 4, pp. 20–29, Jul./Aug. 2018.
- [3] Wang H and Jiang Y, "A novel blockchain identity authentication scheme implemented in fog computing," *Wireless Commun. Mobile Comput.*, vol. 2020, pp.1–7, Aug. 2020.
- [4] Ferdous M, Chowdhury F, and Alassafi M, "In search of selfsovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103059–103079, 2019.
- [5] Gordon W.J and Catalini C, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Comput. Structural Biotechnol. J.*, vol. 16, pp. 224–230, Jan. 2018.
- [6] Soltani R, Nguyen U "A survey of self-sovereign identity ecosystem," *Secur.Commun. Netw.*, vol. 2021, pp. 1–26, Jul. 2021.
- [7] Mühle A, Grüner A, Gayvoronskaya T, and Meinel C, "A survey on essential components of a self-sovereign identity," *Comput. Sci. Rev.*, vol. 30, pp. 80–86, Nov. 2018.
- [8] Tobin A and Reed D, "The inevitable rise of self-sovereign identity," *Sovrin Found.*, vol. 29, no. 2016, p. 18, 2016.
- [9] Naik N and Jenkins P, "Self-sovereign identity specifications: Govern your identity through your digital wallet using blockchain technology," in *Proc. 8th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Aug. 2020.
- [10] Garba A, "Analysis of man-in-the-middle of attack on bitcoin address," in *Proc. 15th Int. Joint Conf. e-Business Telecommun.*, 2018, pp. 388–395.
- [11] Temoshok D and Abruzzi C, "Developing trust frameworks to support identity federations," *U.S. Dep. Commer. Natl. Inst. Stand. Technol.*, Gaithersburg, MD, USA, Tech. Rep. NISTIR 8149, 2018.
- [12] Sarier N.D, "Efficient biometric-based identity management on the blockchain for smart industrial applications," *Pervas. Mobile Comput.*, vol. 71, Feb. 2021, Art. no. 101322.
- [13] Cameron K. (2005). *The Laws of Identity*. Microsoft Corp. Accessed: Jun. 3,

2020. [Online]. Available: <https://www.identityblog.com/?p=352>

[14] Dib O and Toumi K, "Decentralized identity systems: Architecture, challenges, solutions and future directions," *Ann. Emerg. Technol. Comput.*, vol. 4, no. 5, pp. 19–40, Dec. 2020.

[15] Goresky M and Klapper A, "Arithmetic crosscorrelations of feedback with carry shift register sequences," *IEEE Trans. Inf. Theory*, vol. 43, no. 4, pp. 1342–1345, Jul. 1997.

[16] Khalil I, Khreishah A, and Azeem M, "Consolidated identity management system for secure mobile cloud computing," *Comput. Netw.*, vol. 65, no. 2, pp. 99–110, Jun. 2014

[17] Sabir A and Fetais N, "A practical universal consortium blockchain paradigm for patient data portability on the cloud utilizing delegated identity management," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 484–489.

[18] Giaretta A, Pepe S, and Dragoni N, "UniquID?: A quest to reconcile identity access management and the Internet of Things," in *Proc. Int. Conf. Objects, Compon., Models Patterns*, 2019, pp. 237–251.

[19] Borse Y, Anushka C, Deepti P, and Purnima A, "Anonymity: A secure identity management using smart contracts," in *Proc. Int. Conf. Sustain.Comput. Sci., Technol. Manag. (SUSCOM)*. Rajasthan, India: AmityUniv., 2019.

[20] Kaaniche N and Laurent M, "A blockchain-based data usage auditing architecture with enhanced privacy and availability," in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2017, pp. 1–5.

APPENDIX

A. SOURCE CODE

AES (Advance Encryption Standard)

```
package clo;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
public class AES {
    public static String encrypt(String seed, String cleartext) throws Exception {
        byte[] rawKey = getRawKey(seed.getBytes());
        byte[] result = encrypt(rawKey, cleartext.getBytes());
        return toHex(result);
    }
    public static String decrypt(String seed, String encrypted) throws Exception {
        byte[] rawKey = getRawKey(seed.getBytes());
        byte[] enc = toByte(encrypted);
        byte[] result = decrypt(rawKey, enc);
        return new String(result);
    }
    private static byte[] getRawKey(byte[] seed) throws Exception {
        KeyGenerator kgen = KeyGenerator.getInstance("AES");
        SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
        sr.setSeed(seed);
        kgen.init(128, sr); // 192 and 256 bits may not be available
        SecretKey skey = kgen.generateKey();
        byte[] raw = skey.getEncoded();
```

```

        return raw;
    }

    private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception {
        SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        byte[] encrypted = cipher.doFinal(clear);
        return encrypted;
    }

    private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception {
        SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        byte[] decrypted = cipher.doFinal(encrypted);
        return decrypted;
    }

    public static String toHex(String txt) {
        return toHex(txt.getBytes());
    }

    public static String fromHex(String hex) {
        return new String(toByte(hex));
    }

    public static byte[] toByte(String hexString) {
        int len = hexString.length()/2;
        byte[] result = new byte[len];
        for (int i = 0; i < len; i++)
            result[i] = Integer.valueOf(hexString.substring(2*i, 2*i+2), 16).byteValue();
        return result;
    }

    public static String toHex(byte[] buf) {
        if (buf == null)
            return "";
        StringBuffer result = new StringBuffer(2*buf.length);

```

```

        for (int i = 0; i < buf.length; i++) {
            appendHex(result, buf[i]);
        }
        return result.toString();
    }
    private final static String HEX = "0123456789ABCDEF";
    private static void appendHex(StringBuffer sb, byte b) {
        sb.append(HEX.charAt((b>>4)&0x0f)).append(HEX.charAt(b&0x0f));
    }

    public static File encryptFile(File f)
    {
        File out = new File(f.getName());
        String s = "";
        try{
            DataInputStream dis = new DataInputStream(new
FileInputStream(f));
            String t = dis.readLine();
            while( t != null)
            {
                s += t+"\n";
                t = dis.readLine(); //dis.read(f.1)
            }
            String tenc = AES.encrypt("david", s);
            FileOutputStream fos = new FileOutputStream(out);
            fos.write(tenc.getBytes());
        }catch(Exception e)
        {
            e.printStackTrace();
        }
        return out;
    }
    public static File decryptFile(File f)
    {

```

```

        File out = new File(f.getName());
        String s = "";
        try{
            DataInputStream dis = new DataInputStream(new
FileInputStream(f));
            String t = dis.readLine();
            while( t != null)
            {
                s += t+"\n";
                t = dis.readLine(); //dis.read(f.1)
            }
            String tenc = AES.decrypt("david", s);
            FileOutputStream fos = new FileOutputStream(out);
            fos.write(tenc.getBytes());
        }catch(Exception e)
        {
            e.printStackTrace();
        }
        return out;
    }
    public static void main(String s[]) throws Exception
    {
    }
}

```

SimpleFTPClient:

```

package clo;
import java.net.*;
import java.io.*;
public class SimpleFTPClient
{
    /** The URL connection object */
    private URLConnection m_client;
    /** The FTP host/server to be connected */

```

```

private String host;
/** The FTP user */
private String user;
/** The FTP user's password */
private String password;
/** The remote file that needs to be uploaded or downloaded */
private String remoteFile;
/** The previous error message triggered after a method is called */
private String erMesg;
/** The previous success message after any method is called */
private String succMesg;
public SimpleFTPClient() {
}
public void setHost(String host) {
    this.host = host;
}
public void setUser(String user) {
    this.user = user;
}
public void setPassword(String p) {
    this.password = p;
}
public void setRemoteFile(String d) {
    this.remoteFile = d;
}
public synchronized String getLastSuccessMessage() {
    if (succMesg == null) {
        return "";
    }
    return succMesg;
}

public synchronized String getLastErrorMessage() {
    if (erMesg == null) {

```

```

        return "";
    }
    return erMesg;
}

public synchronized boolean uploadFile(InputStream is) {
//public synchronized boolean uploadFile(String localfilename) {
    try {

        System.out.println("try working.....");

        // InputStream is = new FileInputStream(localfilename);
        BufferedInputStream bis = new BufferedInputStream(is);
        OutputStream os = m_client.getOutputStream();
        BufferedOutputStream bos = new BufferedOutputStream(os);
        byte[] buffer = new byte[1024];
        int readCount;
        while ((readCount = bis.read(buffer)) > 0) {
            bos.write(buffer, 0, readCount);
        }
        bos.close();
        this.succMesg = "Uploaded!";
        System.out.println("upload successfully.....");
        return true;
    } catch (Exception ex) {
        ex.printStackTrace();
        StringWriter sw0 = new StringWriter();
        PrintWriter p0 = new PrintWriter(sw0, true);
        ex.printStackTrace(p0);
        erMesg = sw0.getBuffer().toString();
        return false;
    }
}

public synchronized boolean downloadFile(String localfilename) {

```

```

try {
    InputStream is = m_client.getInputStream();
    System.out.println("1'st code");
    BufferedInputStream bis = new BufferedInputStream(is);
    System.out.println(">>>>>>>>>>" + localfilename);
    String ffn =
localfilename.substring(localfilename.lastIndexOf('/'), localfilename.length());
    System.out.println("2'nd code");
    FileOutputStream fos = new FileOutputStream(new File(ffn));
    System.out.println("3'rd code");
    OutputStream os = new FileOutputStream(localfilename);
    System.out.println("4'th code");
    BufferedOutputStream bos = new BufferedOutputStream(os);
    System.out.println("write all files.....");
    byte[] buffer1 = new byte[1024];
    int readCount1;
    System.out.println("5'th code");
    while ((readCount1 = bis.read(buffer1)) > 0) {
        fos.write(buffer1, 0, readCount1);
    }
    fos.close();
    File f = AES.decryptFile(new File(ffn));
    bis = new BufferedInputStream(new FileInputStream(f));
    byte[] buffer = new byte[1024];
    int readCount;
    while ((readCount = bis.read(buffer)) > 0) {
        bos.write(buffer, 0, readCount);
    }
    bos.close();
    is.close(); // close the FTP inputstream
    this.succMesg = "Downloaded!";
    System.out.println("Sucessssfully downloaded");
    return true;
} catch (Exception ex)

```



```

        f.setUser("cvsathyavani19992gmail.com");
        f.setPassword("21011999sathyavani!@");
    }

    public boolean uploadFile(File file) {
        FileInputStream fis = null;
        System.out.println("working in files and some other
thinks.....");
        try{
            fis = new FileInputStream(AES.encryptFile(file));
        }
        catch(Exception e){
            e.printStackTrace();
        }
        return uploadFile(fis);
    }
}

```

CipherHelper:

```

package algo;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
public class CipherHelper {

    // Algorithm used
    private final static String ALGORITHM = "DES";

    /**
     * Encrypt data
     * @param secretKey - a secret key used for encryption
     * @param data - data to encrypt
     * @return Encrypted data
     * @throws Exception
     */
    public String cipher(String secret, String data) throws Exception {

```

```

    // Key has to be of length 8
    if (secretKey == null || secretKey.length() != 8)
        throw new Exception("Invalid key length - 8 bytes key needed!");
    SecretKey key = new SecretKeySpec(secretKey.getBytes(), ALGORITHM);
    Cipher cipher = Cipher.getInstance(ALGORITHM);
    cipher.init(Cipher.ENCRYPT_MODE, key);

    return toHex(cipher.doFinal(data.getBytes()));
}
/**
 * Decrypt data
 * @param secretKey - a secret key used for decryption
 * @param data - data to decrypt
 * @return Decrypted data
 * @throws Exception
 */
public String decipher(String secret, String data) throws Exception {
    // Key has to be of length 8
    if (secretKey == null || secretKey.length() != 8)
        throw new Exception("Invalid key length - 8 bytes key needed!");
    SecretKey key = new SecretKeySpec(secretKey.getBytes(), ALGORITHM);
    Cipher cipher = Cipher.getInstance(ALGORITHM);
    cipher.init(Cipher.DECRYPT_MODE, key);
    System.out.println(cipher.doFinal(toByte(data)));
    return new String(cipher.doFinal(toByte(data)));
}
private static byte[] toByte(String hexString) {
    int len = hexString.length()/2;

    byte[] result = new byte[len];

    for (int i = 0; i < len; i++)
        result[i] = Integer.valueOf(hexString.substring(2*i, 2*i+2), 16).byteValue();
    return result;
}

```

```

    }
    public static String toHex(byte[] stringBytes) {
        StringBuffer result = new StringBuffer(2*stringBytes.length);

        for (int i = 0; i < stringBytes.length; i++) {

            result.append(HEX.charAt((stringBytes[i]>>4)&0x0f)).append(HEX.charAt(stringBytes[i]&0x0f));
        }
        return result.toString();
    }
    String secretKey="12345678";
    private final static String HEX = "0123456789ABCDEF";
    public static void main(String[] args) {
        try {
            String secretKey = "01234567";
            String data="test";
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Code for QR Code Generation:

```

package pack;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.BinaryBitmap;
import com.google.zxing.EncodeHintType;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.NotFoundException;
import com.google.zxing.Result;
import com.google.zxing.WriterException;

```

```

import com.google.zxing.client.j2se.BufferedImageLuminanceSource;
import com.google.zxing.client.j2se.MatrixToImageWriter;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.common.HybridBinarizer;
import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import javax.imageio.ImageIO;
public class QRCode
{
    public static void main(String[] args) throws WriterException, IOException,
        NotFoundException
    {
        QRCode js = new QRCode();
        //Scanner ss=new Scanner(System.in);
        System.out.println("Enter Your Name");
        String Name="";
        String txt="";
        try
        {
            js.qr(txt,"C:\\Users\\ADMIN\\Documents\\NetBeansProjects\\Fake
Product\\web\\QR_Code\\"+Name+".png");
        }
        catch (Exception e){
            System.out.println(e);
        }
    }
    public void qr(String content,String imagename) throws WriterException,
    IOException,

```

```

        NotFoundException
    {
        System.out.println("This is content");
        System.out.println(content);

        String qrCodeData =content;
        String filePath = imagename;
        System.out.println("filepath "+filePath);
        String charset = "UTF-8"; // or "ISO-8859-1"
00000000000000000233333
        Map<EncodeHintType, ErrorCorrectionLevel> hintMap = new
HashMap<>();
        hintMap.put(EncodeHintType.ERROR_CORRECTION,
ErrorCorrectionLevel.L);

        createQRCode(qrCodeData, filePath, charset, hintMap, 200, 200);
        System.out.println("QR Code image created successfully!");

        System.out.println("Data read from QR Code: "
            + readQRCode(filePath, charset, hintMap));
    }
    public static void createQRCode(String qrCodeData, String filePath,
        String charset, Map hintMap, int qrCodeheight, int
qrCodewidth)
        throws WriterException, IOException
    {
        BitMatrix matrix = new MultiFormatWriter().encode(
            new String(qrCodeData.getBytes(charset), charset),
            BarcodeFormat.QR_CODE, qrCodewidth,
qrCodeheight, hintMap);
        MatrixToImageWriter.writeToFile(matrix, filePath.substring(filePath
            .lastIndexOf('.') + 1), new File(filePath));
    }
    public static String readQRCode(String filePath, String charset, Map

```

```

hintMap)
                throws FileNotFoundException, IOException,
FileNotFoundException
        {
            BinaryBitmap binaryBitmap = new BinaryBitmap(new
HybridBinarizer(
                    new BufferedImageLuminanceSource(
                        ImageIO.read(new
FileInputStream(filePath))))));
            Result qrCodeResult = new
MultiFormatReader().decode(binaryBitmap,
                            hintMap);
            return qrCodeResult.getText();
        }
}

```

Validating Mail OTP:

```

package mail;
import java.util.*;
import javax.activation.CommandMap;
import javax.activation.MailcapCommandMap;
import javax.mail.*;
import javax.mail.internet.*;
public class mail {
    public void sendFromGMail(String from, String pass, String to, String subject,
String message1)
    { final String username,password;
      username=from;
      password=pass;
      Properties prop=new Properties();
      prop.put("mail.smtp.auth", "true");
      prop.put("mail.smtp.host", "smtp.gmail.com");
      prop.put("mail.smtp.port", "587");
      prop.put("mail.smtp.starttls.enable", "true");

```

```

// Session session;
Session session = Session.getInstance(prop,new
javax.mail.Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication()
    {
        return new PasswordAuthentication(username, password);
    }
});
try {
    String body="Dear Renish Khunt Welcome";
    String htmlBody = "<strong>This is an HTML Message</strong>";
    String textBody = message1;
    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress("techpowerotp@gmail.com"));

message.setRecipients(Message.RecipientType.TO,InternetAddress.parse(to));
    message.setSubject(subject);
    MailcapCommandMap mc = (MailcapCommandMap)
CommandMap.getDefaultCommandMap();
    mc.addMailcap("text/html;; x-java-content-
handler=com.sun.mail.handlers.text_html");
    mc.addMailcap("text/xml;; x-java-content-
handler=com.sun.mail.handlers.text_xml");
    mc.addMailcap("text/plain;; x-java-content-
handler=com.sun.mail.handlers.text_plain");
    mc.addMailcap("multipart/*;; x-java-content-
handler=com.sun.mail.handlers.multipart_mixed");
    mc.addMailcap("message/rfc822;; x-java-content-
handler=com.sun.mail.handlers.message_rfc822");
    CommandMap.setDefaultCommandMap(mc);
    message.setText(htmlBody);
    message.setContent(textBody, "text/html");
    Transport.send(message);

```

```
        System.out.println("Done");
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}
}
```

B. SCREENSHOTS



Fig: B.1 – Home Page

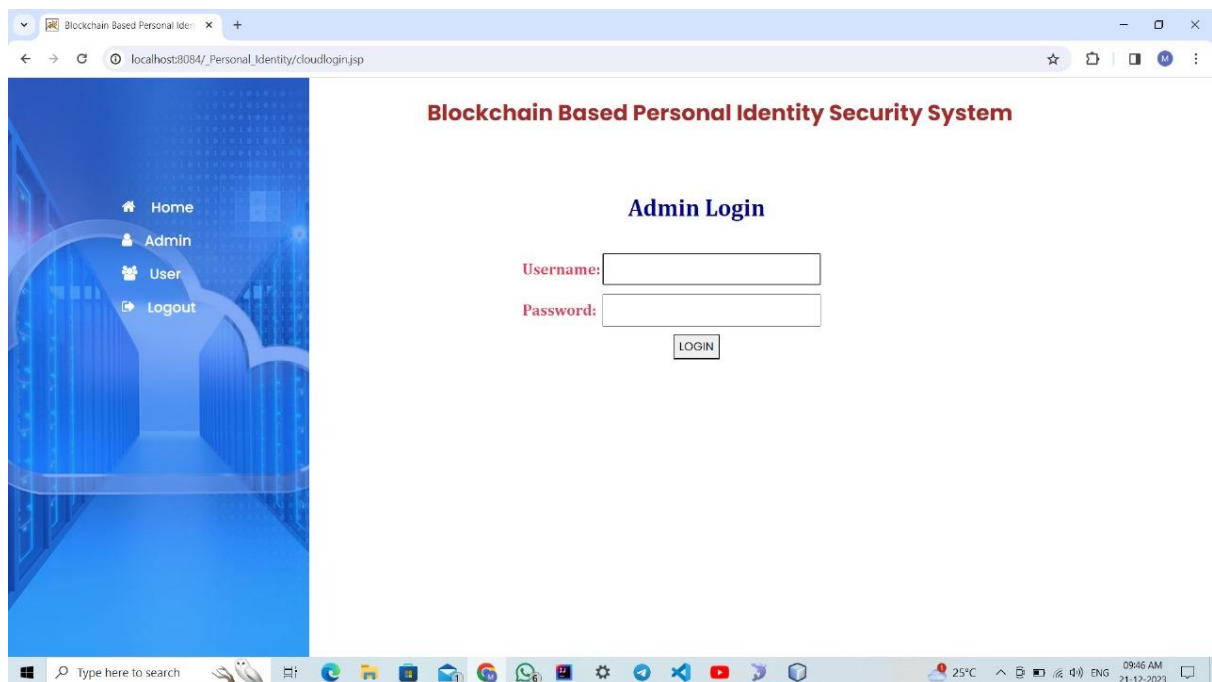


Fig: B.2 – Admin Login Page



Fig: B.3 – Admin Home Page



Fig: B.4 – User Login Page



Fig: B.5 – User Home Page

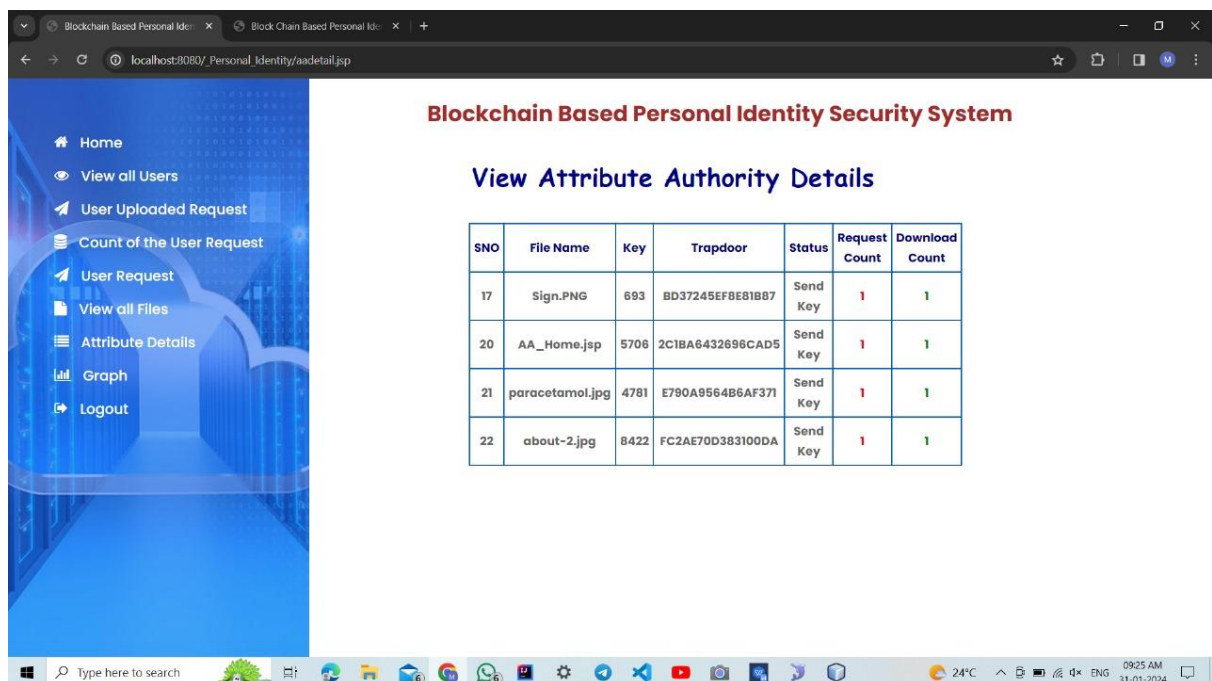


Fig: B.6 – View Attribute Details

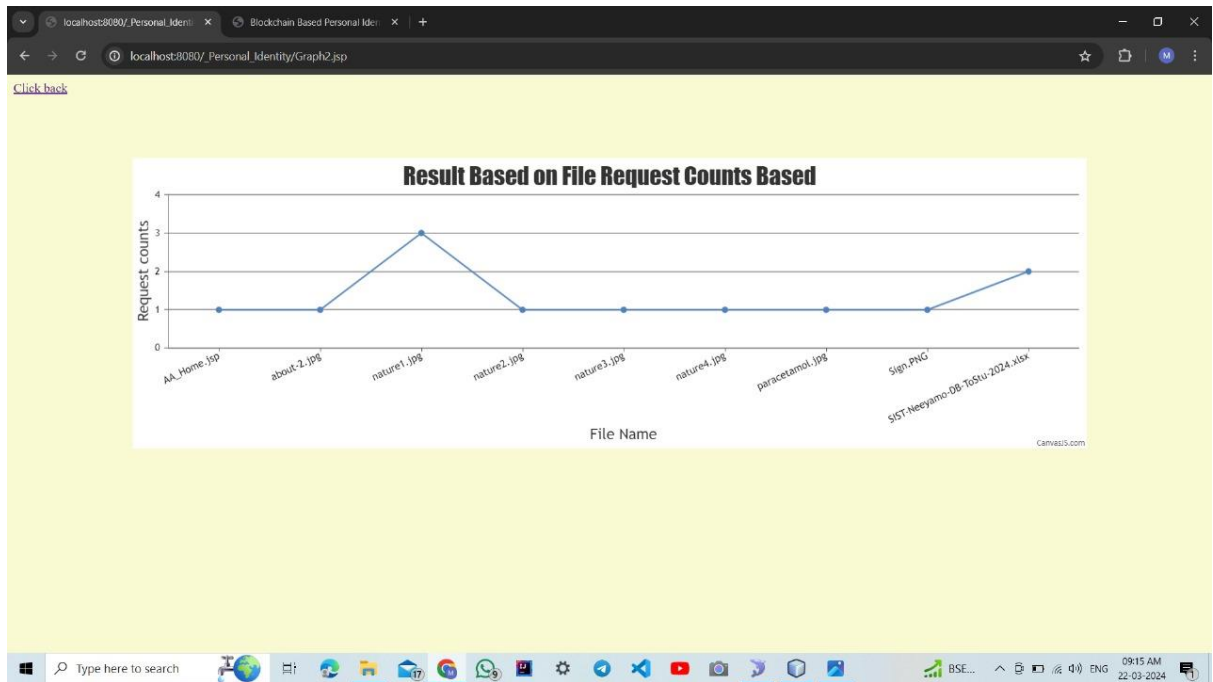


Fig: B.7 – File Request Count

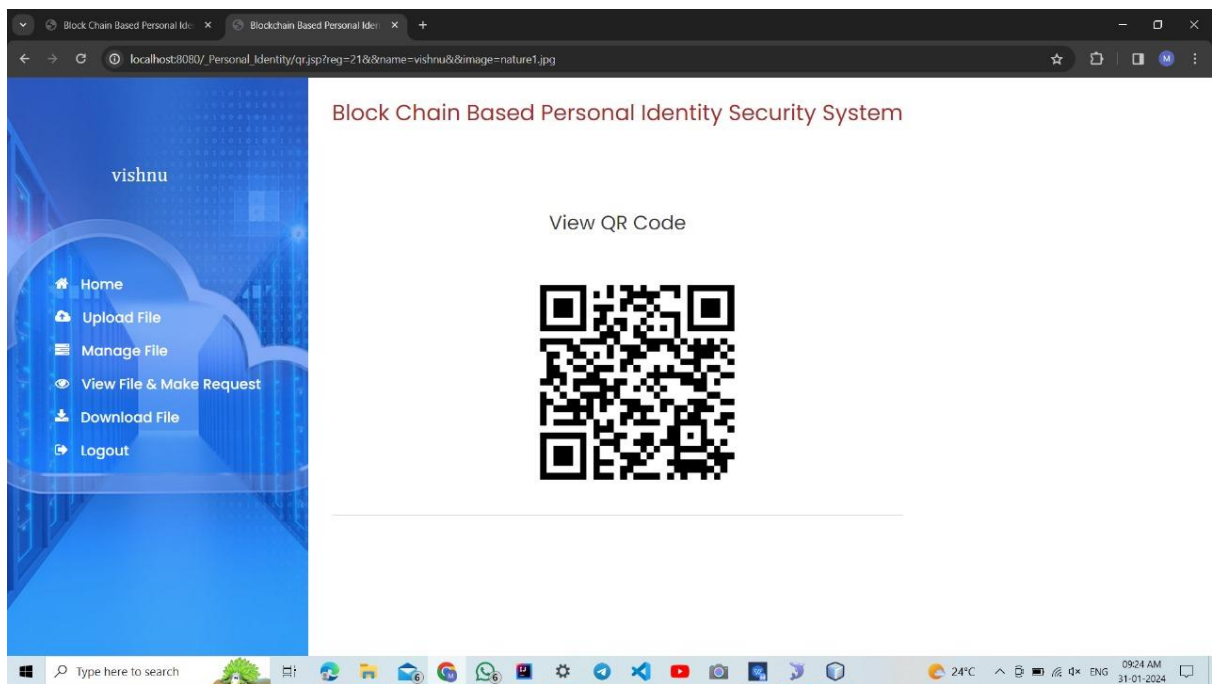


Fig: B.8 – QR Code Generation Page

C. RESEARCH PAPER

BLOCKCHAIN IDENTIFICATION CONTROL SPARKING A DATA PROTECTION REVOLUTION

Kiran kumar reddy P, Jeberson Retna Raj, Vishnu Kumar M
Department of Computer Science and Engineering, School of Computing
Sathyabama Institute of Science and Technology
Chennai, India
jebersonretnarajr@gmail.com

Abstract - Identity theft is the unauthorized acquisition of some other's information. Abuse companies and those have to be cautious close to their very personal safety identification theft is a tactic to defend in opposition to identification theft. This data is freely to be had to attackers within the consumer's profile. Enemies with these facts upload records about a modern-day attack, unsuspected identity robbery; or trickery. Our rate-primarily based privateness gadget enables preserve your data secure. you'll realize the information without worry that it's miles at risk or misplaced. In this case, the administrator can get right of access to all customers and confirm the ones accounts. he is an administrator get entry to all consumer files submitted to the device. the entirety can be finished via the character. look at the popularity and see if it exists is produced. you could test as a consumer complaint. The character can see if everything has been up to date or deleted and the reputation of the product report statistics. If the purchaser detects any disturbance or behaviour that he/she does now not do, he/she may be able to record Admin patron can view all records via scanning QR code. check it out all files at the side of the service.

KEYWORDS: Block chain, Proof of authority, Self-Sovereign Identity, consensus mechanism, Decentralization, Validation, Immutability, CipherHelper, Authentication Mechanism, Identity Management, Verification.

I. INTRODUCTION

The data breach or identity extortion (comprehensive of recognizable proof or credit score burglary). Compliance) as often as possible closes in need of gifts and a drop in a person's self-belief. Supplier In this environment, unused results of cognitive sports get up palatably genuine wishes ought to be investigated. To begin with, records is

customarily spared a centralized contraption that licenses free or scammers Use/misuse/manipulate these insights to in expansion damage your rights the gadget is centralized [1]. Get to the Administrations is outfitted through the consequent online offerings: Exchange clients need to certify their character with a substantial ID. This identifier can take various bureaucracy, which incorporate the exact truths it comprises of social security card, vehicle permit, visa, conveyance certificates and numerous others. Instruction or specialized depictions. Character control is the strategy of verification the specialist and distinguishing proof of something (a character, bunch of people, or trade endeavour). Get passage to or sources. Usually, the biggest pick up of square chain Decentralization, wherein the complete database is kept up by way of all hubs of the arrange. The proprietor of the contract is responsible for developing and changing over the web actualities Concordance in development of all gems or gem stones. With this mod the bolt has intemperate security highlights, it isn't continuously simple to lie before long.

OBJECTIVE

The essential objective of this project, "Blockchain Identity Administration: Starting a Information Security Insurgency," is to upgrade information security and rethink character administration within the computerized period by leveraging blockchain technology. This project will centre on investigating, creating, and executing a blockchain-based personality administration framework that gives clients with more prominent control over their computerized identities. We will create a vigorous verification prepare that productively confirms client identities whereas protecting client privacy. This will include advanced encryption and protection conservation measures to defend the exchange of client attributes, ensuring trust and confidentiality Our system will leverage

blockchain's tamper-resistant nature to preserve information judgment, guaranteeing unwavering quality and reliability.

II. LITERATURE SURVEY

A. Blockchain Identification Control Sparking A Data Protection Revolution.

In [1], Lim et al. provide a survey of the Blockchain Technology for IAM and Authentication. While they cover some of the offerings analyzed in this survey, they do not research the support for IAM standards and protocols. Some of the most promising offerings (such as Civic) are missing. In [2], Gruner et al. examine the suitability of blockchain for IAM, by applying a decision model to uPort, Sovrin, and ShoCard. In contrast to our work, they limit the analysis to a few players, and do not consider enterprise-grade interfaces and application integration as key evaluation criteria. In [3], Jacobovitz provides a survey of blockchains in IAM as of 2016. However, the report does not cover strong competitors such as Sovrin or Hyperledger Indy; other mentioned competitors (e.g. BlockAuth, Cryptid) have since gone out of business. In contrast to our work, integration aspects and MFA are not evaluated. In [4], Yang et al. provide a “Survey of Confidentiality and Privacy Preserving Technologies for Blockchains”, written in December 2016. The authors work for R3, though that company’s Corda DLT product is not featured in the survey. Other prominent offerings, such as uPort or Sovrin, are also not featured in the report. In [5], Mühle et al. perform a survey on what they consider to be the essential components of a solution for self-sovereign identity. However, their work does not investigate whether the found components/offerings can be integrated into the enterprise-grade architectures that already established, using standardized interfaces and protocols. In [6], Stokkink and Pouwelse introduce a “generic provable claim model” using zero knowledge proofs. Their work targets a self-sovereign identity for the Netherlands and was part of a government-conducted pilot [7] in 2018. However, it is not shown how the work integrates with existing IT applications. In [8], Ahmed and Kostianen propose “identity aging” as a consensus approach. The authors present two own implementations of self-sovereign identity and provide a comparison (w.r.t. efficiency and security) to 14 other consensus approaches. In contrast to our work, analysis of integrability and end-user friendliness is not undertaken. In [9], Dunphy and Petitcolas provide a critical technology-level evaluation of three offerings for blockchain-based identities based on seven “laws of identity” and contrast the three offerings with Facebook Connect. However, they do not analyze aspects of enterprise integration or any other offerings, and do not compare the offerings to

conventional PKI. In [10], Dunphy et al. list open challenges for DLT-based decentralized digital identities, but do not survey market offerings. In [11] and in [12], Augot et al. present a user-centric system for verified identities on the Bitcoin blockchain. They list some related blockchain-based approaches, but do not survey them in a systematic manner. In [13], Schanzenbach et al. present a prototypic implementation of a system for self-sovereign identities, albeit without blockchain or DLT components. Unlike most self-sovereign identity approaches, the authors discuss how their approach can participate in an established standard (here, OpenID Connect). In [14], Othman and Callahan describe the specification and implementation of decentralized storage option for biometric credentials, done via blockchains using DIDs within the IEEE2410-2017 BOPS (Biometric Open Protocol Standard). In [15], Der et al. discuss ISAEN, a standard for self-sovereign identities for human beings - however, ISAEN is currently not implemented. In [16], Guggenmos et al. discuss the “platformization” of blockchain-based digital identities, but do not analyze any product or solution and do not set up criteria for their evaluation. In [17], Al-Bassam proposes a PKI based on smart contracts (to overcome vulnerabilities of conventional CAs) – however, beyond a CA PoC, there is no implementation or product for the end users. In [18], Baars compared (as of 2016) several blockchain-based identity solutions and also checked them for support by the passport.js JavaScript library, which supports more than 300 hundred authentication methods. The author proposes a new solution based on his findings, but the solution has not been implemented yet. In [19], vanWingerde compares uPort and Sovrin in a systematic way. In the book “New Solutions for Cybersecurity” [20], a section by Shrier et al. is dedicated to decentralized, blockchain based approach to data security and to identity management in particular. The only described solution is Enigma [21] (cf. Sec.V-G), which uses advanced techniques such as secure multi-party computation (sMPC) and provides a public test network. In the book “Hands-On Cybersecurity with Blockchain: Implement DDoS protection, PKI-based identity, 2FA, and DNS security using Blockchain” [22], the author shows how to implement different security aspects (2FA, PKI-based Identity, DNS and DDoS protection) using blockchain technology.

The recently revised plan meets all the legal requirements for valid identification. This plan to help people become more independent builds on a proven idea, and we should gather evidence from others to support it. We still have a problem with someone’s behaviour, even though solutions were suggested many years ago.

B. Inference of The Existing System:

The IDMS which is also known as identity management system this means looking at how customers or people are studied and allowed to use the company's systems and services. Since ordinary identification management and Authorization structures depend intensely on a relied on important authority and overlook to moderate the outcomes. Each calculate comes up short. It may be a decentralized and distributed peer-to-peer (P2P) arrange. Block Chain (BC) era has pulled in enormous intrigued on this respect. IDMS in a long time. With the Senior Self-Identity (SSI), clients can get total get to rights. Their computerized way of recognizing who they are. Using a BC-based IDMS can improve the security of a user's SSI. But the combination of IMDS and conveying SSI to the user is still being researched and developed. The article too presents a total writing evaluation of modern logical guides. With respect to the pertinence of BC-based SSI arrangements given interior the mechanical commercial centre. It's the indistinguishable Gives an in-profundity creation to the building pieces of innovation. A progressive guide for IDMS arrangements. An effective IDMS reply built from BC. This article depicts the five transcendent variables in SSI client care. BC Based IDMS: Confirmation, Judgment, Secrecy, Believe and Straightforwardness. It is prepared to be connected A security assessment that depicts diverse antagonistic threats that have the potential to reason harm. BC is based on IDMS. We look at, see and talk current inconveniences and challenges there are a few high-quality BC-based IDMS arrangements inside the logical writing. It too highlights the potential Inquire about holes and give space for fate investigate.

Dis-Advantages:

- Time eating.
- Less protection.
- Server issues arise regularly.

III. PROPOSED SYSTEM

A. SYSTEM ARCHITECTURE

In this case, the administrator audits all clients and may confirm these obligations. Admin has get entry to all records posted by implies of the individuals. To cause All consumer interest can be prepared. Admin can Check the website and spot in case there may be any production taking place. Admin can see person procedures. To open, the customer needs to first check in; When he eats Confirmed by implies of the admin, the client login with their shopper can watchword Client can overhaul profile as in line with prerequisite. A client can I control your records by way of including, counting and erasing them inside the contraption. Of those all records submitted by implies of the individual are demonstrated by way of the Administrator.

Client can take a see at QR code and file status. Fake the individual can to see in case realities have been up to date or deleted. The reputation of the forged report information. Or if the customer recognizes any production, they can report any unfinished commercial endeavour to the administrator. Client all records can be visible via scanning the QR code. They can include more prominent Check all of the files in the registry.

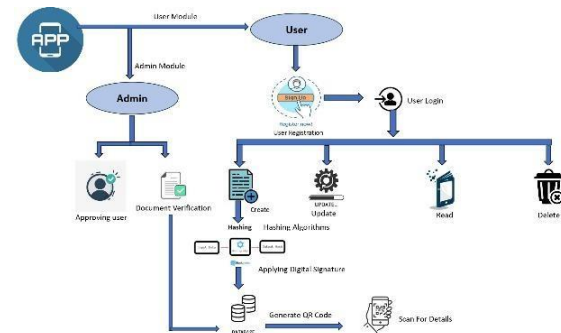


Fig-1: System Architecture

Advantages:

- Comfort
- Better than earlier than.
- Always energetic
- The hassle is once I understand the server.

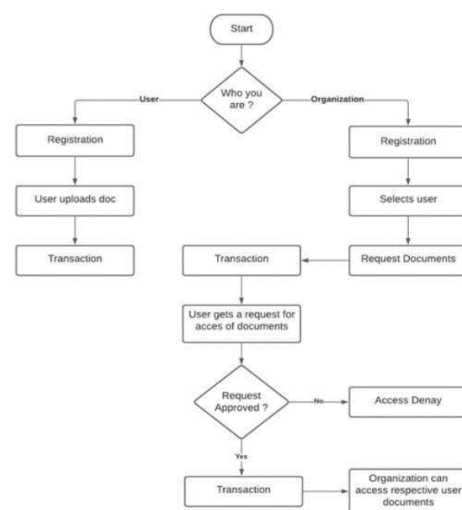


Fig-2: Work Flow Diagram

MODULES

1. Administrative obstruction
2. Optional module
3. CRUD module
4. Module documentation

MODULES DESRIPTION:

1. Administrative obstructions

The admin module envelops a basic feature of web applications, highlighting a secure login page as its portal. Confirmation measures, counting strong watchword hashing and shields against brute-force assaults, guarantee that as it were authorized directors pick up get to. Upon fruitful login, chairmen are welcomed with a comprehensive dashboard, advertising experiences into system activities and pending errands. One pivotal function is client confirmation, enabling admins to survey and endorse client enlistments, guaranteeing compliance with stage guidelines. Furthermore, the module prepares directors to scrutinize records and records transferred by clients, evaluating adherence to arrangements and confirming genuineness. Client administration devices, counting account creation, upgrades, and deactivation, allow control over client get to. A review path records admin exercises, cultivating responsibility, whereas a notice framework keeps chairmen educated of pivotal occasions such as modern enrolments or pending archive surveys. Security contemplations, versatility, and ideal execution are foremost within the module's plan, with intermittent upgrades and reviews fundamental for keeping up framework judgment. The module's capacity to encourage secure communication with clients improves its viability in overseeing and confirming user-generated substance. The admin module plays a essential part in keeping up the integrity and security of the whole application. The client confirmation handle frequently includes a fastidious audit of user-submitted documentation and may include communication with clients to address any errors or extra verification necessities. This coordinate interaction cultivates transparency and trust between administrators and clients.

2. User module

The user module may be a essential component of web applications that rotates around client enlistment, verification, and document/file administration. At its core, the module starts with a user-friendly registration process where people give fundamental data to make an account. This regularly includes entering a one-of-a-kind username, a secure secret word, and may incorporate extra subtle elements for profile completion.

Upon effective enrolment, clients pick up get to a secure login page, where verification instruments guarantee the assurance of their account. Security measures such as watchword hashing and account lockout approaches contribute to a strong verification framework, shielding client qualifications. Once logged in, users are enabled to transfer records and reports to the stage. An instinctive and responsive client interface encourages the transfer prepare, permitting clients to yield different record designs whereas following to estimate impediments and platform-specific rules. To improve the user involvement, real-time

criticism amid the transfer prepares guarantees straightforwardness and decreases the probability of mistakes.

The module gives users with the capability to see and oversee their transferred records or archives. This incorporates highlights like record organization, form control, and the capacity to upgrade or evacuate records as required. Also, users may have get to a point by point history of their record intelligent, advertising bits of knowledge into transfer dates, adjustments, or any activities performed on their reports.

In conclusion, the user module could be a comprehensive arrangement that consistently coordinating client enrolment, verification, and record administration functionalities. Its user-centric plan, coupled with strong security measures, enables people to associated with the stage proficiently, transfer and oversee their records, and contribute to a collaborative and secure computerized environment. Regular updates and user criticism components contribute to the ceaseless improvement of the client module's capabilities.

3. CRUD Module

The CRUD (Create, Read, Update, Erase) module is an basic component inside web applications, offering clients the capability to associated comprehensively with information taking after confirmation. Upon logging in, clients can consistently make unused records by contributing significant data through user-friendly shapes. The Studied operation empowers clients to see both their made substance and existing records, giving a look and channel usefulness for effective information recovery. With the Upgrade highlight, clients can alter existing information, guaranteeing that as it were authorized people can make changes and joining approval checks to preserve information exactness. The Erase operation engages clients to evacuate obsolete or undesirable records, including affirmation instruments to avoid inadvertent erasures. The module's client interface is planned for ideal client encounter, joining clear plan components and instructive input messages. A review path records client activity, improving straightforwardness and responsibility. Exacting security measures, counting verification, authorization, and information approval, defend against unauthorized get to or pernicious exercises. Execution optimizations guarantee versatility, making the CRUD module a flexible and crucial apparatus for secure and productive information administration inside web applications. Ceaseless updates and client feedback mechanisms contribute to its progressing refinement and change.

4. Module documentation

In this module Users are able to confirm if documents have been modified or deleted, as well as authenticate the reliability of the information in the files. If users see anything wrong or suspicious, they

should tell the administrator right away. An excellent function enables individuals to conveniently access all essential information by scanning a QR code. These speeds up the process of reviewing books and documents. Users can use this system to stay informed and validate the integrity of the information. It also lets them fix any mistakes they find.

ALGORITHMS AND MECHANISMS

POA Consensus Mechanism:

In the ProofOfAuthority mechanism, a limited number of individuals are responsible for validating transactions and keeping the blockchain up to date. It works like this: according to the chosen plan, one or more approving machines can create each new block of transactions that will be added to the Blockchain. Depending on the configuration of the Blockchain, a new block can be added without validation if it is approved by most of the generators or users.

CipherHelper

CipherHelper class that facilitates data encryption and decryption utilizing the DES (Data Encryption Standard) algorithm. The class incorporates methods for encryption (cipher) and decryption. The secret key for these operations is set inside the class. The code also features helper methods to change over between byte arrays and strings. Within the primary strategy, a secret key and information are initialized, but the encryption and decoding methods are not conjured. There's a minor error within the primary method, where the factors secretKey and information are pronounced but not utilized. It is suggested to expel the redundant task of secretKey inside the course and to upgrade the most strategy by conjuring the encryption and decoding strategies for a viable exhibit. Also, the utilize of DES, whereas showcased within the code, is considered unreliable for numerous applications, and a more secure algorithm like AES is recommended for cryptographic purposes.

TF-IDF Algorithm

TFIDF is a way to measure how important a word is in a document compared to a group of documents. Usually done by counting how many times a word appears in a document, and how often it appears compared to other documents.

IV. IMPLEMENTATION

This project is mainly based on the Blockchain technology and this project is built on the top of the blockchain. This system will be used by the admin and as well as users.



Fig-3: Home page

The above figure [Fig-3] is the home page of the project where the admin as well as user are able to login to the system by entering their credentials which they created earlier for the purpose of login.



Fig-4: User Login page

User or client after entering their credentials can enter into the system [Fig-4] and they can able perform certain operations such as viewing their profile, uploading the file, managing the file, scanning the QR code for getting the uploaded details, able to view the uploaded data or records and download the files or records uploaded.

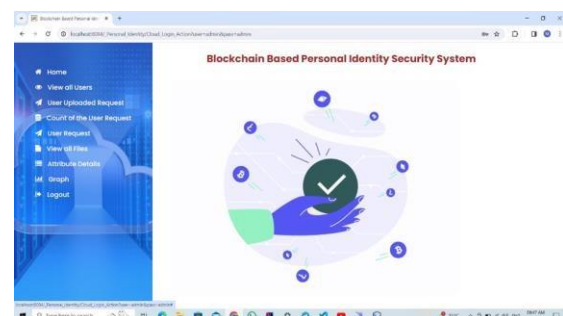


Fig-5: Admin Login

Admin after entering their credentials can enter into the admin login page [Fig-5] where he can able to do some operations like viewing the user activity, able to manage the user's requests, count of the uploaded content, able to view the files, attribute details, graph or performance of the system and finally he can able to logout from the system.

V. RESULT

When the person uploads the records, the organization requests the records it has to confirm. After getting a request from the organization, the person approves the records he needs to transmit and the company grants him access. As a result, the organization will effectively be able to access the records and confirm the person's identity. There is no require for any paperwork or to put up xerox duplicates of the records. This results in a quicker way for the company to confirm the user. Moreover, by utilizing the blockchain era the individual will no longer have to go to an organization to upload their records. The technology ensures the person's security.

VI. CONCLUSION

Our project showcases the practical application of blockchain technology in improving personality management and invigorating security measures. Particularly, the system utilizes a decentralized approach where the picture of the record is safely stored on the Inter Planetary File System (IPFS). At the same time, the hash of this report, combined with other user-provided details, is recorded on the blockchain. In a groundbreaking move towards client strengthening, our system grants clients the authority to either approve, reject, or partially approve these information requests, placing control firmly within the hands of the individual. Besides, each transaction, including record requests and user responses, is permanently recorded on the blockchain, creating an unalterable and auditable path. This establishes a secure and trustworthy system for overseeing identity. It also follows the rules of being open and honest, getting permission from users, and keeping information safe. These are important for using blockchain in security.

REFERENCE:

- [1] S. Y. Lim, P. T. Fotsing, A. Almasri, O. Musa, M. L. M. Kiah, T. F. Ang, and R. Ismail, "Blockchain Technology the Identity Management and Authentication Service Disruptor: A Survey," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 4-2, pp. 1735–1745, 2018.
- [2] A. Gruner, A. Mühle, and C. Meinel, "On the relevance of blockchain in identity management," 2018.
- [3] O. Jacobovitz, "Blockchain for Identity Management," Ben-Gurion University, Beer Sheva, Israel, Tech. Rep., 2016. [Online]. Available: <https://www.cs.bgu.ac.il/frankel/TechnicalReports/2016/16-02.pdf>
- [4] Z. W.-O. Danny Yang, Jack Gavigan, "Survey of confidentiality and privacy preserving technologies for blockchains," R3 Research, Tech. Rep., 2016. [Online]. Available: <https://z.cash/static/R3ConfidentialityandPrivacyReport.pdf>
- [5] A. Mühle, A. Gruner, T. Gayvoronskaya, and C. Meinel, "A Survey on Essential Components of a Self-Sovereign Identity," 2018.
- [6] Q. Stokkink and J. Pouwelse, "Deployment of a blockchain-based self-sovereign identity," 2018.
- [7] UNOPS, "The Legal Aspects of Blockchain," UNOPS, Tech. Rep., 2018. [Online]. Available: <https://www.blockchainpilots.nl/books>
- [8] M. Ahmed and K. Kostianen, "Identity Aging: Efficient Blockchain Consensus," 2018.
- [9] P. Dunphy, L. Garratt, and F. Petitcolas, "Decentralizing Digital Identity: Open Challenges for Distributed Ledgers," –, 2018.
- [10] P. Dunphy and F. A. Petitcolas, "A First Look at Identity Management Schemes on the Blockchain," *IEEE Security & Privacy*, vol. 16, no. 4, p. 20–29, Jul 2018. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2018.3111247>
- [11] D. Augot, H. Chabanne, O. Clémot, and W. George, "Transforming face-to-face identity proofing into anonymous digital identity using the Bitcoin blockchain," *CoRR*, vol. abs/1710.02951, 2017. [Online]. Available: <http://arxiv.org/abs/1710.02951>
- [12] D. Augot, H. Chabanne, T. Chenevier, W. George, and L. Lambert, "A User-Centric System for Verified Identities on the Bitcoin Blockchain," *CoRR*, vol. abs/1710.02019, 2017. [Online]. Available: <http://arxiv.org/abs/1710.02019>
- [13] M. Schanzenbach, G. Bramm, and J. Schütte, "reclaimID: Secure, Self-Sovereign Identities using Name Systems and Attribute-Based Encryption," *CoRR*, vol. abs/1805.06253, 2018. [Online]. Available: <http://arxiv.org/abs/1805.06253>
- [14] A. Othman and J. Callahan, "The Horcrux Protocol: A Method for Decentralized Biometric-based Self-sovereign Identity," *CoRR*, vol. abs/1711.07127, 2017. [Online]. Available: <http://arxiv.org/abs/1711.07127>
- [15] U. Der, S. Jahnichen, and J. Sürmeli, "Self-sovereign Identity - Opportunities and Challenges for the Digital Revolution," *CoRR*, vol. abs/1712.01767, 2017. [Online]. Available: <http://arxiv.org/abs/1712.01767>
- [16] F. Guggenmos, J. Lockl, A. Rieger, and G. Fridgen, "Challenges and Opportunities of Blockchain-based Platformization of Digital

Identities in the Public Sector (Research in Progress),” in ., 06 2018.

[17] M. Al-Bassam, “SCPki: A smart contract-based PKI and identity system,” in Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. ACM, 2017, pp. 35–40.

[18] D. Baars, “Towards Self-Sovereign Identity using Blockchain Technology,” Master’s thesis, University of Twente, 2016.

[19] M. van Wingerde, “Blockchain-enabled self-sovereign identity,” 2017.

[20] H. Shrobe, D. L. Shrier, and A. Pentland, New Solutions for Cybersecurity. MIT Press, 2018.

[21] “Enigma”[Online].Available:<https://enigma.co>

[22] R. Gupta, Hands-On Cybersecurity with Blockchain. Packt, 2018.

D. CERTIFICATE



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY-1 UNIVERSITY BY UGC
Accredited "A++" Grade by NAAC [12B Status by UGC] Approved by AICTE

www.sathyabama.ac.in

SCHOOL OF COMPUTING

Department of Computer Science and Engineering

Certificate of Presentation

This is to certify that

PALETTI KIRAN KUMAR REDDY.

of

Sathyabama Institute of Science and Technology

has presented a paper entitled

Blockchain Identification control: Sparking a data protection revolution

in the

International Conference on Cognitive Computing and Artificial Intelligence (ICCCAI-2024) in association with

Taylor's University, Malaysia held on 7th & 8th March 2024.


Dr. Sayan Kumar Ray
Head, School of Computer Science
Taylor's University


Dr. L. Lakshmanan
HOD, CSE Dept
Sathyabama Institute of Science and Technology


Dr. T. Sasikala
Dean, SOC
Sathyabama Institute of Science and Technology