

SECURE FILE SHARING SYSTEM USING CRYPTOGRAPHY AND BLOCKCHAIN

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in INFORMATION TECHNOLOGY

By

RAUSHAN KUMAR (Reg No. 39120092)
DIGANTA PANDA (Reg No. 39120024)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

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

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

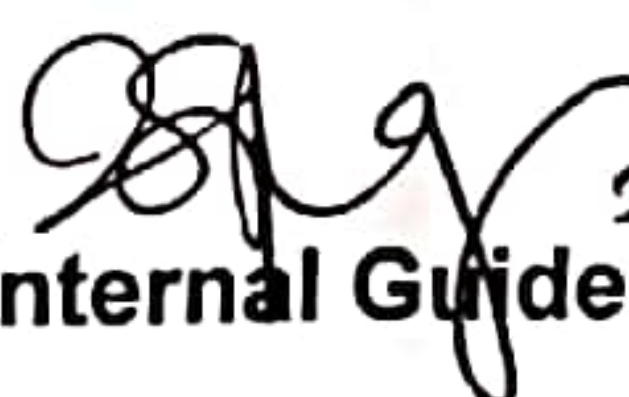
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DIGANTA PANDA** (Reg.No-39120024) and **RAUSHAN KUMAR** (Reg.No-39120092) who carried out the Project Phase-2 entitled "SECURE FILE SHARING SYSTEM USING CRYPTOGRAPHY AND BLOCKCHAIN" under my supervision from January 2023 to May 2023.

 21/April/2023
Internal Guide

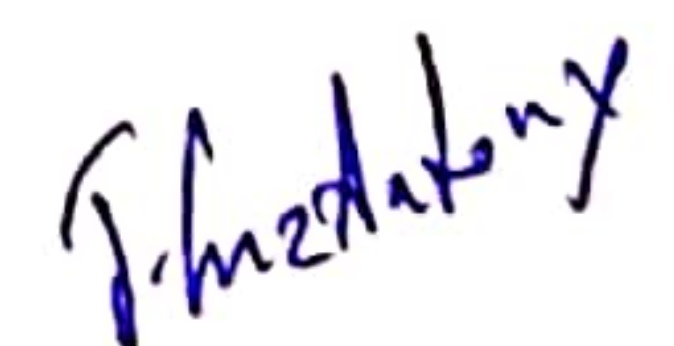
Dr.G. MATHIVANAN, M.E, Ph.D.,


Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 10/5/23


Internal Examiner


External Examiner

DECLARATION

I, DIGANTA PANDA (Reg.No-39120024), hereby declare that the Project Phase -2 Report entitled "SECURE FILE SHARING SYSTEM USING CRYPTOGRAPHY AND BLOCKCHAIN" done by me under the guidance of Dr. G. MATHIVANAN, M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 10/5/23

PLACE: Chennai

Diganta



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 kind 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, **Dr. R SUBHASHINI M.E., Ph.D.**, Head of the Department of Information Technology and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G MATHIVANAN M.E., Ph.D.**, for his valuable guidance, suggestions and constantly encouraging me for the success full completion of my phase-2 project work.

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

ABSTRACT

In today's world data is the most important thing. From small enterprises to trillion-dollar multinational companies, data is the most important resource and is worth billions of dollars. So, its security and privacy become essential and uncompromisable. But nowadays hackers and spammers find vulnerabilities and steal the private data and misuse it by selling them in dark web etc. Some employees also share valuable data through WhatsApp which is very risky. To deal this, our paper proposes a system which uses combination of blockchain and cryptography to make those highly safe and secured and strictly prohibits any third or unwanted party from accessing the data. It uses SHA 256 algorithm for cryptographic security. It uses AES algorithm for encryption-decryption. It uses symmetric key for privacy. We have added another layer of security by implementing QR code scanning. Ultimately, staff members, team leaders and higher officials like managers or senior executives can share data securely. No hacker can breach the system. Our work can be deals, financial transactions, online receipt, interpersonal lives of ordinary people thus making it realistic and very practical.

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		viii
1	INTRODUCTION		1-2
2	LITERATURE SURVEY		3-12
	2.1 Literature Survey		3-11
	2.2 Inferences from Literature Survey		12
3	REQUIREMENTS ANALYSIS		13-22
	3.1	Feasibility Studies	13
	3.2	Risk Management	14-15
	3.3	Hardware/Software Requirements Specification document	15-22
4	DESCRIPTION OF PROPOSED SYSTEM		23-41
	4.1	Selected Methodology or process model	23-28
	4.2	Architecture / Overall Design of Proposed System	29-32
	4.3	Modules	33-37
	4.4	Project Management Plan	37-39
	4.5	Transition/ Software to Operations Plan	40-41
5	IMPLEMENTATION DETAILS		42-49
	5.1	Development and Deployment Setup	42-45
	5.2	Algorithms	45-48
	5.3	Testing	49
6	RESULTS AND DISCUSSION		50
7	CONCLUSION		51-53
	7.1	Conclusion	51
	7.2	Future work	51
	7.3	Research Issues	52
	7.4	Implementation Issues	53
	REFERENCES		54-55

	APPENDIX	
	A. SCREENSHOTS	56-59
	B. SOURCE CODES	59-71
	C. RESEARCH PAPER	72-78

LIST OF ABBREVIATION

S.No	ABBREVIATION	EXPANSION
1	DB	Database
2	SMC	Secure Multiparty Computation
3	MDA	Medical Admin
4	DBC	Data Base Confidentiality
5	JVM	Java Virtual Machine
6	JSP	Java Server Page

LIST OF FIGURES

FIG No.	DESCRIPTION	PAGE No.
4.1	System Architecture Diagram	29
4.2	Data Flow Diagram	30-31
4.3	ER Diagram	32
4.4	MODULE DIAGRAM	35-37

CHAPTER 1

INTRODUCTION

Sensitive data transaction is confidential information that must be kept safe and out of reach from all outsiders unless they have permission to access it. Access to sensitive data should be limited through sufficient data security and information security practices designed to prevent data leaks and data breaches. Sensitive data can be any sort of information that needs to be protected from unauthorized access to safeguard the privacy or security of an individual or organization.

Recently, a variety of efficient encryption algorithms has been proposed in the cloud computing framework, and these algorithms can provide good privacy protection functions for uploaded data. Among them, the ciphertext-policy attribute-based encryption (CP-ABE) technology has become a very attractive topic due to its fine-grained access control, one-to-many data sharing, high security, and other advantages. Based on the identity-based encryption technology, proposed the first CP-ABE encryption primitive, which designs an access control tree to be associated with the ciphertext. Each non leaf node can be expressed as a threshold gate each leaf node can be associated with an attribute.

The first CP-ABE prototype was proposed by Bethencourt et al. Then, many schemes implemented CP-ABE schemes with multiple authorization centers, such as and in order to solve the key management issues. Meanwhile, some schemes have been given for improving the efficiency of CP-ABE technology, or hiding the access structure. However, these schemes can deal with the situation that a single user performs encryption and decryption operations.

A framework of CP-ABE technology is briefly introduced as follows. 1) Setup CP-ABE: The system setup algorithm inputs the secure parameter λ , and then, it outputs the public parameter PK and the master key MSK. Note that MSK is saved locally and PK is the implicit input of each following algorithms. 2) Key Gen CP-ABE: The attribute-key generation algorithm inputs the master key MSK and the attribute set S. Then, it returns the attribute-key SK to the user. 3) Encrypt CP-ABE: The DO runs the encryption algorithm to encrypt the plaintext m with the access structure.

File sharing is the practice of distributing or providing access to digital media, such as computer programs, multimedia (audio, images and video), documents or electronic books. Common methods of storage, transmission and dispersion include removable media, centralized servers on computer networks, Internet-based hyperlinked documents, and the use of distributed peer-to-peer networking. File sharing technologies, such as BitTorrent, are integral to modern media piracy, as well as the sharing of scientific data and other free content.

Data sharing is one of the core functions of every business. Every organization, regardless of the industry it belongs to, requires a reliable data-sharing platform. The crucial process of decision-making in businesses across the globe greatly depends on the data shared. But with the rise in online transactions and information access, the traditional Web 2.0 data sharing methodologies are increasingly proving to be both costly and unreliable.

The process of data sharing for organizations requires to be transformed to provide higher reliability and security. One of the solutions to the challenges of sharing data over centralized databases includes using blockchain technology. In this article, we shall discuss the role of blockchain in data sharing and how it can help organizations make informed decisions based on unaltered data.

Blockchain is a digital ledger of transactions that get duplicated and distributed over the entire network. It offers a decentralized system for managing data and transactions in a peer-to-peer network. Unlike the traditional way of sharing data through clouds in Web 2.0, the decentralized system allows the distribution of data divided into blocks. Each block is then protected from intruders attempting to alter the data. Every change in the data needs to be validated by all peers or miners (users) in the network using advanced cryptography techniques.

Data sharing in blockchain can provide organizations with a secure way to store and distribute data. Given the decentralized nature of this network, there are no owners in the blockchain database, and the data travels through the peer-to-peer network secured by an immutable cryptographic signature. Blockchain helps make the data difficult to hack or change.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

1. TITLE: CNN Based Malicious Website Detection by Invalidating Multiple Web Spams.

AUTHOR: Dongjie Liu And Jong-Hyouk Lee

YEAR: 2020

PAPER EXPLANATION:

Although a variety of techniques to detect malicious websites have been proposed, it becomes more and more difficult for those methods to provide a satisfying result nowadays. Many malicious websites can still escape detection with various Web spam techniques. In this paper, we have summarized three types of Web spam techniques used by malicious websites, such as redirection spam, hidden Iframe spam, and content hiding spam. We then present a new detection method that adopts the perspective of users and takes screenshots of malicious webpages to invalidate Web spams. The proposed detection method uses a Convolutional Neural Network, which is a class of deep neural networks, as a classification algorithm. In order to verify the effectiveness of the method, two different experiments have been conducted. First, the proposed method was tested based on a constructed complex dataset. We present comparison results between the proposed method and representative machine learning-based detection algorithms. Second, the proposed method was tested to detect malicious websites in a real-world Web environment for three months. These experimental results illustrate that the proposed method has a better performance and is applicable to a practical Web environment.

2. TITLE: A systematic literature review of blockchain cyber security

AUTHOR: Paul J. Taylor, Tooska Dargahi , Ali Dehghantanha, Reza M. Parizi, Kim-Kwang Raymond Choo.

YEAR: 2020

PAPER EXPLANATION:

Since the publication of Satoshi Nakamoto's white paper on Bitcoin in 2008, blockchain has (slowly) become one of the most frequently discussed methods for securing data storage and transfer through decentralized, trustless, peer-to-peer systems. Our findings show that the Internet of Things (IoT) lends itself well to novel blockchain applications, as do networks and machine visualization, public key cryptography, web applications, certification schemes and the secure storage of Personally Identifiable Information (PII). This timely systematic review also sheds light on future directions of research, education and practices in the blockchain and cyber security space, such as security of blockchain in IoT, security of blockchain for AI data, and sidechain security etc.

3. TITLE: Secure Computation by Secret Sharing using Input Encrypted with Random Number

AUTHOR: Keiic Keiichi Iwamura and Ahmad Akmal Aminuddin Mohd Kamalhi
Iwamura and Ahmad Akmal Aminuddin Mohd Kamal

YEAR: 2019

PAPER EXPLANATION:

Typically, unconditionally secure computation using a (k, n) threshold secret sharing is considered impossible when $n < 2k - 1$. Therefore, in our previous work, we first took the approach of finding the conditions required for secure computation under the setting of $n < 2k - 1$ and showed that secure computation using a (k, n) threshold secret sharing can be realized with a semi-honest adversary under the following three preconditions: (1) the result of secure computation does not include 0; (2) random numbers reconstructed by each server are fixed; and (3) each server holds random numbers unknown to the adversary and holds shares of random numbers that make up the random numbers

unknown to the adversary. In this paper, we show that by leaving condition (3), secure computation with information-theoretic security against a semi-honest adversary is possible with $k \leq n < 2k$ –

1. In addition, we clarify the advantage of using secret information that has been encrypted with a random number as input to secure computation. One of the advantages is the acceleration of the computation time. Namely, we divide the computation process into a preprocessing phase and an online phase and shift the cost of communication to the preprocessing phase. Thus, for computations such as inner product operations, we realize a faster online phase, compared with conventional methods.

4. TITLE: Secure multi-party computation in differential private data with Data Integrity Protection

AUTHOR: Sundari Ananthi_M

YEAR: 2015

PAPER EXPLANATION:

Secure multiparty computation (SMC) is needed now-a-days in which data are distributed between different parties. Moreover, organizations are wished to collaborate with other parties who conduct same business, for their mutual benefits. SMC provides users to gain much information from the larger data without disclosing the data. This project combines the technique secure multi-party computation and the differential privacy for vertically partitioned data between parties. To achieve this, a multi-party protocol has been proposed for the exponential mechanism. Reliable access to data is must for most computer applications and data servers. Some factors cause unauthorized access to stored data. Two Phase Validation (2PV) provides the authentication for the users, while integrating the data in multiparty computation. Data can get corrupted due to some malfunctions. Disk errors are common today but the storage technologies are not designed to handle such kind of errors. A simple integrity violation is detected by the higher-level software which causes further loss of data. The proposed system is to verify the integrity of random subsets of data against general or malicious corruptions through Distributed Data Integrity (DDI) Protection.

5. TITLE: A Ciphertext-Policy Attribute-Based Encryption Based on an Ordered Binary Decision Diagram.

AUTHOR: Long Li, Tianlong GU and Yenning Liu

YEAR: 2017

PAPER EXPLANATION:

Ciphertext-policy attribute-based encryption (CP-ABE) is widely used in many cyber physical systems and the Internet of Things for guaranteeing information security. In order to improve the performance and efficiency of CP-ABE, this paper makes a change to the access structure of describing access policies in CP-ABE, and presents a new CP-ABE system based on the ordered binary decision diagram (OBDD). The new system makes full use of both the powerful description ability and the high calculating efficiency of OBDD. First, in the access structure, the new system allows multiple occurrences of the same attribute in a strategy, supports both positive attribute and negative attribute in the description of access policies, and can describe free-form access policies by using Boolean operations. Second, in the key generation stage, the size of secret keys generated by the new system is constant and not affected by the number of attributes; furthermore, time complexity of the key generation algorithm is $O(1)$. Third, in the encryption stage, both the time complexity of the encryption algorithm and the size of generated ciphertext are determined by the number of valid paths contained in the OBDD instead of the number of attributes occurring in access policies. Finally, in the decryption stage, the new system supports fast decryption and the time complexity of the decryption algorithm is only $O(1)$. As a result, compared with existing CP-ABE schemes, the new system has better performance and efficiency. It is proved that the new CP-ABE system can also resist collision attack and chosen-plaintext attack under the decisional bilinear Diffie Hellman assumption.

6. TITLE: White-Box Traceable Ciphertext-Policy Attribute-Based Encryption in Multi-Domain Environment

AUTHOR: Yongli Tang, Xu He, Yi Chang Shek

YEAR: 2020

PAPER EXPLANATION:

In a ciphertext-policy attribute-based encryption (CP-ABE) system, the decryption keys are only related to attributes shared by multiple users and do not contain any identity information of their original holders. Hence, if a decryption key is leaked, there is no feasible method to trace the suspicious user. The user tracing problem has become an obstacle to the adoption of CP-ABE in practice. In order to address it, some traceable/accountable CP-ABE schemes have been established. However, considering the user tracing problem in multi-domain environment will face new challenges. Multi-domain environments usually have a two-layer structure, domains and intradomain users. When tracing a user, we should first trace the domain where the user is located, and then trace the user in the domain. Unfortunately, the existing traceable CP-ABE schemes only focus on one level of users tracing, and are not suitable for the two-layer structure of multi-domain environments. Therefore, a white-box traceable CP-ABE scheme in multi-domain environment is proposed in this paper. The proposed scheme obtains a so-called two-layer tracing. At the domain level, a short signature technique is used to prevent an attacker from forging the tracing parameter and realize the traceability for domains. Linkable ring signature technology is introduced at the user level to provide a tracing method for users by utilizing the likability of the signature. The two signature structures are reasonably embedded in a user private key to support two-layer white-box tracing at both domains and intradomain users. The proposed scheme supports any monotone access structures and has full security against chosen plaintext attack in the standard model. Compared with related schemes, the asymptotic communication cost and the asymptotic computation cost of the proposed scheme are relatively low. These advantages make the proposed scheme more practical for solving the user tracing problem in multi-domain environment.

7. TITLE: Updatable Ciphertext-Policy Attribute-Based Encryption Scheme With Traceability and Revocability

AUTHOR: Jing Xu, Yan Liu

YEAR: 2020

PAPER EXPLANATION:

Ciphertext-policy attribute-based encryption (CP-ABE) can offer fine-grained access control over encrypted data, which is suitable for complex commercial applications. However, since the same decryption privileges could be shared by multiple users in the one-to-many encryption mechanism, it is dangerous that a malicious user misuses his secret key but cannot be traced. In addition to further security, when the malicious user has been caught, it is required to revoke him from the system. To address these problems, we propose a novel updatable CP-ABE scheme supporting white-box traceability and traitor revocation. In the proposed scheme, a “fixed point” is embedded into the user's secret key to achieving the traceability and each user is assigned with a unique identifier for revocation. Moreover, the secret exponent used to encrypt a message is divided into two parts: one is assigned to access policy and the other to the revocation list. Therefore, only a part of the ciphertext components needs to be updated when the revocation list is changed, which greatly simplifies the process of ciphertext update. Compared to the previous works, our scheme is more efficient, and can achieve valid revocation and ciphertext update. In addition, the traceability of the proposed scheme is depended on the ℓ -Strong Diffie-Hellman assumption, and the indistinguishability security under selective access policy and chosen-plaintext attacks in the standard model is reduced to the Decisional q -Bilinear Diffie-Hellman assumption. Furthermore, the experimental results show that the proposed scheme is efficient.

8. TITLE: Ciphertext-Policy Hierarchical Attribute-Based Encryption Against Key-Delegation Abuse for IoT-Connected Healthcare System

AUTHOR: Xi Chen, Yong Li And Yun Chao

YEAR: 2020

PAPER EXPLANATION:

With the maturity of technologies such as Internet of Things (IoT) and Cloud Computing,

more and more medical institutions share patient's medical data in the IoT-connected healthcare system, whereas it also brings hidden dangers to data security and privacy protection. The application of Hierarchical Attribute-Based Encryption (HABE) in IoT-connected healthcare system helps to solve the problem of data sharing with a large number of users. Delegation is a function of HABE, which can realize the transfer of hierarchical user access rights, effectively reducing the workload of the trusted authority. However, when "delegation" is used to generate decryption keys for users who do not comply with the access structure, the "key-delegation abuse" problem arises, which can seriously damage the privacy of patient's data. Nevertheless, it has not attracted much attention in previous research. This paper proposes the problem of key-delegation abuse in Ciphertext-Policy Hierarchical Attribute-Based Encryption (CP-HABE). After the cryptanalysis of typical schemes, we conclude two reasons for key-delegation abuse in CP-HABE, which are randomizing the original key elements or keeping parts of them in a new decryption key. This paper proposes a new mechanism specifically by using directed graph and construct a CP-HABE scheme against key-delegation abuse (CP-HABE-AKDA). Aiming at the scenario of key leaking, we further present the traceable CP-HABE-AKDA solution for the IoT-connected healthcare system, which has the additional function of tracking and verifying the identity of key leaker.

9. TITLE: An Encrypted Cloud Email Searching and Filtering Scheme Based on Hidden Policy Ciphertext-Policy Attribute-Based Encryption With Keyword Search

AUTHOR: Dongjie Liu And Jong-Hyouk Lee

YEAR: 2021

PAPER EXPLANATION:

With the rapid growth of cloud email services, email encryption is beginning to be used more and more to alleviate concerns about cloud privacy and security. However, this increase in usage invites the problem of how to search and filter encrypted emails effectively. Searchable public key encryption is a popular technology to solve encrypted email searching, but encrypted email filtering is still an open problem. We propose an

encrypted cloud email searching and filtering scheme based on hidden policy ciphertext-policy attribute-based encryption with keyword search as a new solution. It enables the recipient to search the encrypted cloud email keywords and allows the email filtering server to filter the encrypted email content when receiving the email, as the traditional email keyword filtering service. Our hidden policy scheme is constructed by composite order bilinear groups and proven secure by dual system encryption methodology. Our scheme can be applied to other scenarios such as file searching and filtering and has certain practical value.

10.TITLE: A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing

AUTHOR: Guefeng Lin And Hanshu Hong

YEAR: 2017

PAPER EXPLANATION:

Ciphertext policy attribute-based encryption (CP-ABE) is a promising cryptographic technique for fine-grained access control of outsourced data in the cloud. However, some drawbacks of key management hinder the popularity of its application. One drawback in urgent need of solution is the key escrow problem. We indicate that front-end devices of clients like smart phones generally have limited privacy protection, so if private keys are entirely held by them, clients risk key exposure that is hardly noticed but inherently existed in previous research. Furthermore, enormous client decryption overhead limits the practical use of ABE. In this paper, we propose a collaborative key management protocol in CP-ABE. Our construction realizes distributed generation, issue and storage of private keys without adding any extra infrastructure. A fine-grained and immediate attribute revocation is provided for key update. Our encryption technique has a polycentric attribute authorization system based on an AND gate access structure, with a unified attribute index established by each attribute authority throughout the system, to form a binary tree, i.e., attribute access tree. The state value of the parent node can be determined by the state of its child node in an attribute access tree. The proposed collaborative mechanism effectively solves not only key escrow problem but also key exposure. Meanwhile, it helps

markedly reduce client decryption overhead. A comparison with other representative CP-ABE schemes demonstrates that our scheme has somewhat better performance in terms of cloud-based outsourced data sharing on mobile devices. Finally, we provide proof of security for the proposed protocol.

11. TITLE: Efficient Compressed Ciphertext Length Scheme Using Multi-Authority CP-ABE for Hierarchical Attributes

AUTHOR: Zhiyong Zhang And Cheng Li

YEAR: 2018

PAPER EXPLANATION:

In an attribute-based encryption, the user is identified with help of some attributes and their functions for encryption and decryption of the data. The current techniques based on attribute-based encryption have found that if user's access structure includes a considerable amount of attribute information labeled as Don't Care, then the encryption pairing operation has low calculation efficiency and ciphertext information redundancy. In this paper, we have proposed a hierarchical multi-authority attribute-based encryption on prime order groups to tackle these problems. Our encryption technique has a polycentric attribute authorization system based on an AND gate access structure, with a unified attribute index established by each attribute authority throughout the system, to form a binary tree, i.e., attribute access tree. The state value of the parent node can be determined by the state of its child node in an attribute access tree. The attribute-based encryption established in this manner is theoretically proven to effectively decrease the calculation amount for decryption and compress the redundant information in the ciphertext as much as possible.

2.2 INFERENCES FROM LITREATURE SURVEY

- From the above-mentioned literature works, it is evident that there has been surplus research on various.
- It is clear that all of them have their specific pros and cons.
- Few of the researches have used RSA or DES algorithm for encryption which is lesser effective. And some of them have used SHA1 tool whose updated version has arrived.
- Few publications are very secure but few have some drawbacks too by which hackers can break the integrity of the system and can access the secret data.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

3.1.1 *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.

3.1.2 *ECONOMIC FEASIBILITY*

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

3.1.3 *OPERATIONAL 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.

RISK MANAGEMENT

The following risks have been identified, and risk management strategies have been put in place to mitigate their impact on the project:

- **Technical risk:** There is a risk of technical challenges arising during the development and testing phase. To mitigate this risk, regular code reviews, unit testing, and integration testing will be performed.
- **Time risk:** There is a risk of delays in the project timeline due to unforeseen circumstances. To mitigate this risk, contingency plans will be developed, and regular progress reports will be provided to stakeholders.
- **Resource risk:** There is a risk of resource constraints, such as a shortage of skilled developers or hardware failure. To mitigate this risk, contingency plans will be developed, and regular hardware maintenance and backup procedures will be implemented.

Requirement analysis is a critical step in the development of any system, including the proposed two-layer blockchain system for distributed storage in IIoT. Here are some potential requirements for the system:

Scalability: The system should be scalable to handle the large amounts of data generated by IIoT devices. This could be achieved by using a hierarchical structure of storage nodes that can scale up or down easily.

Efficiency: The system should be efficient in terms of computational and energy resources. This could be achieved by using a modified consensus mechanism and lightweight nodes.

Security: The system should ensure the privacy and security of the stored data. This could be achieved by using homomorphic encryption, sharing, and other security mechanisms.

Interoperability: The system should be interoperable with other IIoT systems and devices, allowing for seamless integration and data exchange.

Accessibility: The system should be accessible to authorized users from anywhere at any time, using a user-friendly interface.

Reliability: The system should be reliable, with a high level of uptime and minimal downtime.

Compliance: The system should comply with all applicable laws and regulations, including data protection and privacy laws.

Performance: The system should perform well under heavy loads, with minimal latency and high throughput.

Maintenance: The system should be easy to maintain, with minimal downtime and easy updates.

Cost: The system should be cost-effective, with a reasonable cost of development and implementation.

These are some potential requirements for the proposed two-layer block chain system for distributed storage in IIoT. A comprehensive requirement analysis should be conducted to identify all the specific requirements for the system based on the needs of the target market and stakeholders.

3.3 HARDWARE AND SOFTWARE SPECIFICATION

3.3.1 *HARDWARE REQUIREMENTS*

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

Processor	-	Intel Pentium Dual Core 2.00 GHz
Speed	-	2.40GHz
RAM	-	4GB
Hard Disk	-	20GB

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- FRONTEND : J2EE(JSP,SERVLET), JavaScript, HTML,CSS
- BACKEND : MYSQL 5.5, Java
- OPERATING SYSTEM : WINDOWS 7 or Above
- IDE : ECLIPSE, VS Code, MySQL

JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular

programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

Object-oriented programming allows programmers to break down complex problems into smaller, more manageable pieces by encapsulating data and behavior into objects. This makes the code easier to maintain, modify, and reuse.

Java supports the four main principles of object-oriented programming: encapsulation, inheritance, polymorphism, and abstraction.

Encapsulation is the process of hiding implementation details and exposing only the necessary information to the user. This helps to protect the data and behavior of an object from unwanted access or modification.

Inheritance allows new classes to be based on existing classes, inheriting their data and behavior. This makes it easier to create and maintain code by reusing code that has already been written.

Polymorphism allows objects to take on multiple forms. This means that an object can be used in different contexts and can respond to messages in different ways depending on the context.

Abstraction allows programmers to focus on the essential features of an object, ignoring unnecessary details. This helps to simplify the design of the program and makes it easier to modify and maintain over time.

JSP

Jakarta Server Pages is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, SOAP, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but uses the Java programming language.

One of the main reasons why the Java Server Pages technology has evolved into what it is today and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. Another benefit of utilizing JSP is that it allows to more cleanly separate the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a number of exciting benefits, which are chronicled as follows:

1. The JSP technology is platform independent, in its dynamic web pages, its web servers, and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web-enabled application server. The JSP pages can be accessed from any web server.
2. The JSP technology emphasizes the use of reusable components. These components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the at development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA WebLogic and IBM WebSphere.

HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for its appearance.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img/>` and `<input />` directly introduce content into the page. Other tags such as `<p>` and `</p>` surround and provide information about document text and may include sub-element tags. Browsers do not display the HTML tags but use them to interpret the content of the page.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, which reduces complexity and repetition in the structural content; and enable the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

JAVASCRIPT

JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language. A relational database organizes data

into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, php BB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the

software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License.

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS

4.1.1 BLOCKCHAIN JAVA

Blockchain is a budding technology that has tremendous scope in the coming years. In this tutorial, we will briefly cover the basic concepts of Blockchain. We'll also create a basic Blockchain program in Java to understand how it works in the programming world.

Blockchain is the modern technology that stores data in the form of block data connected through cryptography and cryptocurrencies such as Bitcoin. It was introduced by Stuart Haber and W. Scott Tonette in 1991. It is a linked list where the nodes are the blocks in the Blockchain, and the references are hashes of the previous block in the chain. References are cryptographic hashes when dealing with link lists. The references are just basically objects. So every single node will store another node variable, and it will be the reference to the next node. In this case, the references are cryptographic hashes.

Blockchain uses hash pointers to reference the previous node in a long list. We assign a hash to every single node because this is how we can identify them.

To generate a hash, the SHA256 algorithm is used. One of the most important advantages of this algorithm is that it is generic, so it is a generic cryptographic hash function wherein the input can be anything.

1. Deterministic: It means that if we apply the same hash-function (SHA256) on the same input, the output must be the same.
2. One-way: It is easy to generate the hash with the given hashing algorithm, but on the other hand, it is tough (time-consuming) to restore the original input. It is like a trap-door function.

3. Collision-free: There are no collisions in SHA256 (ok, there are but with extremely low probability). It means that no two different inputs share the same output hash, and this is good. We want to make these hashes unique; this is how we identify a block in the Blockchain.

4. Avalanche effect: A slight change in the input results in a completely different output hash. Otherwise, a cryptanalyst can make predictions about the input based on the output exclusively.

Let's understand the Blockchain concept with the help of a diagram

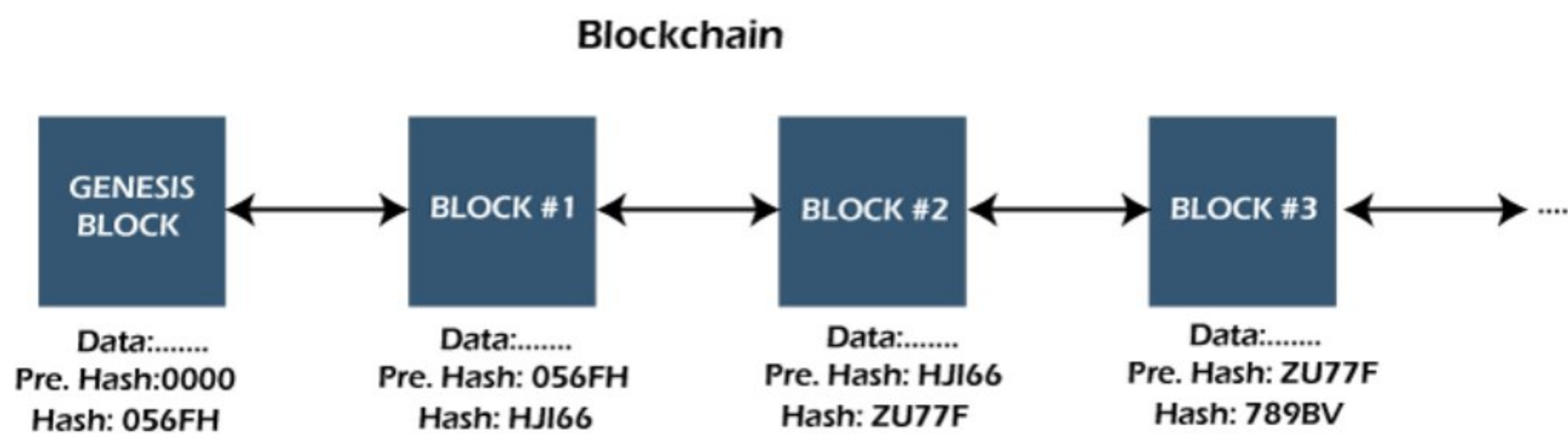


Fig 4.1 Blockchain Block Diagram

- **Genesis Block:** The first block in the blockchain is called the Genesis block. The software constructed it. So, for example, when dealing with Bitcoin cryptocurrency, bitcoin itself created the first block. There is no data associated with this block. The previous hash value is all zeros. And, of course, it has a hash value. For example: Hash = 056FH.
- **Block 1:** The next block in the blockchain will have some data. For example, it could contain the transactions data. The previous hash incorporates the hash value of the Genesis Block. Therefore, the Previous Hash = 056FH
Every node has a hash. So, in this case, the hash of the first block = HJl66
- **Block 2:** The next block is also going to have some data of other transactions. The previous hash value is going to be the hash of Block 1. So, this is why we have been discussing that references are cryptographic hashes. Therefore, the Previous Hash = HJl66
The second block will contain its own hash value. Therefore Hash = ZU77F.

- So basically, this is the underlying data structure. We have various blocks that store data, such as transactions when dealing with cryptocurrencies. Every single block has two hashes, i.e., the hash of the previous node and the hash value of the actual block. And these hash values are going to be the references of the previous blocks implementing Blockchain in Java

Step 1: Defining the constants

Costant.java

STEP 2: Generating Hashes:

In this step, we will implement the SHA256Helper. It is pretty convenient to implement this procedure in Java because we rely on the built methods and built-in classes of Java security.

Firstly, we will import the Message Digest, and further will generate the hash for our Blockchain.

One hash takes up 256 bits in memory with binary values 0 and 1. Therefore, we can conclude that the total number of hashes is 2²⁵⁶. If you are dealing with hexadecimal values, there are 64 hexadecimal characters; therefore, it yields 1664 hash values.

STEP 3: Creating Blocks

The block is the fundamental building block of the Blockchain. These blocks are cryptographically linked together based on the hash values. For creating a block, a Block class is implemented, and further in that class, several variables are defined, such as the id of the block, the timestamp, the hash of the block, the previous hash in the Blockchain, the transactions, and the nonce.

Step 4: Implementing Blockchain

Here we store the generated blocks in an Array List

Step 5: Miner Program

Mining is the most important concept in Blockchain as well as in cryptocurrencies (such as Bitcoin). Mining means finding the right hash to avail you for the given block. But there are some constraints as far as difficulty is concerned. Nevertheless, every miner is going to get a reward because they are validating the given transactions.

In the Constant class, we have already defined the difficulty as '5', which means there have to be five leading zeros at the beginning of every hash. So, miners will generate hash values until they find the right hash.

- Getting rewards is not the only aim of mining; it is just the by-product.
- Mining is the mechanism that allows the Blockchain to be a decentralized security.

- It is about finding the right hash values for the blocks and adding these blocks to the Blockchain.
- Miner is also responsible for appending or adding the block to the Blockchain.

Step 6: Main Program

Now we can run the main blockchain program in java.

4.1.2 **ADVANCED ENCRYPTION STANDARD (AES)**

AES stands for Advanced Encryption Standard and is a majorly used symmetric encryption algorithm. It is mostly utilized to encrypt and safeguard electronic data. AES is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. Because it is faster and more effective than DES (Data encryption standard), it was used to replace DES. Three block cyphers make up the AES encryption system, which is used to encrypt data.

It generates 128 bits of encrypted cypher text as output from 128 bits of input. AES operates utilizing a chain of linked operations that replace and shuffle the input data, which is known as the substitution-permutation network principle. A family of cyphers called Rijndael has various key and block sizes. It is adaptable and has software and hardware implementations. It offers excellent security and can stop many attacks. Since it is copyright-free, anyone can use it anywhere in the world. For 128-bit keys, there are 10 rounds of processing.

There are several **advantages** of using AES algorithm. It can be used with both software and hardware. It offers users a high level of security. It offers one of the top encryptions open-source solutions. This algorithm is very reliable and robust.

Few **disadvantages** are there as well. It requires many rounds for encryption. It is hard to implement on software. It needs much processing at different stages. It is difficult to implement when performance has to be considered.

The AES algorithm's key lengths—128, 192, and 256—have been designed and tested to be strong enough to protect sensitive data higher than or equal to the SECRET level. It will be necessary to use either the 192 or 256 key lengths for TOP SECRET data. Prior to being purchased and used, AES implementation in products meant to safeguard national

security systems and/or information must be examined and certified by the NSA.

4.1.3 HASHING

Hashing is the process of scrubbing raw data until it can no longer be restored to its original form. It takes certain part of information and runs it through a function that performs mathematical operations on the plaintext. This function is known as the hash function, and the output is termed as the hash value.

The hash function converts the plaintext to its respective hash value. It is non-reversible. It means hash value can't be converted back to its original plaintext. No matter how many times they are used, hash functions always produce the same result when the input is left untouched.

4.1.4 SHA 256 (SECURE HASH ALGORITHM)

SHA, which stands for Secure Hash Algorithm, is a family of algorithms that includes SHA 256. In 2001, the NSA and NIST collaborated and introduced it as an advancement for the SHA 1 family, which was gradually becoming less resistant to brute force attacks. The 256 in the name refers to the final hash value, meaning that regardless of the amount of plaintext present, the hash value will always be 256 bits. The other SHA family algorithms are somewhat comparable to SHA 256.

There are certain characteristics of SHA 256 algorithm:

- **Length of the Message:** The cleartext should be no longer than 264 bits. To keep the digest as random as possible, the size must be in the comparison area.
- **Digest Length:** The length of the hash digest for the SHA-256 algorithm should be 256 bits, for the SHA-512 algorithm 512 bits, and so forth. Larger digests typically imply a lot more calculations at the expense of speed and storage.
- **Irreversible:** All hash functions, including the SHA 256, are intended to be irreversible. If you already have the digest, you shouldn't receive the plaintext, and if you run the digest through the hash function once more, it shouldn't return the original value.

Many applications use the SHA algorithm, some of which include the following,

- **digital signature verification:** To confirm the legitimacy of a document or file, digital signatures use asymmetric encryption methodology. To a large extent, hash algorithms like SHA 256 ensure that the signature is verified.

- Password Hashing: Websites store user passwords in a hashed format, for two reasons. Due to the similar size of all the digests, it promotes privacy and lightens the burden on the main database.
- SSL Handshake: Using SHA functions, the SSL handshake is a crucial part of web browsing sessions. To set up a secure connection, your web browsers and the web servers must agree on encryption keys and hash authentication.
- Integrity checks: As was mentioned above, the SHA 256 algorithm and the MD5 algorithm have been used to verify file integrity. It ensures that files were not altered in transit and helps maintain the full functionality of files.

4.2 ARCHITECTURE/OVERALL DESIGN OF PROPOSED SYSTEM

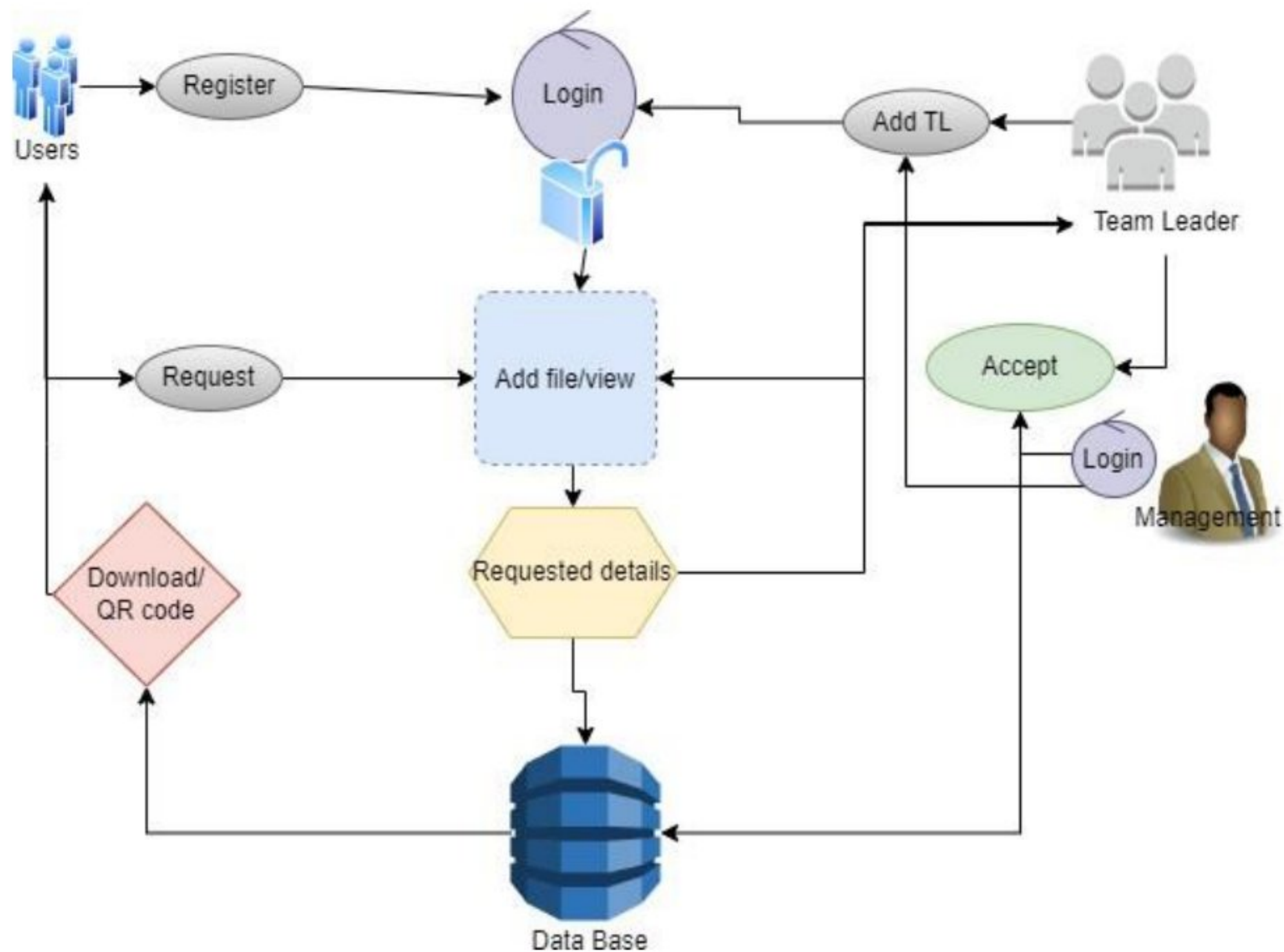


Fig 4.2 System Architecture Diagram

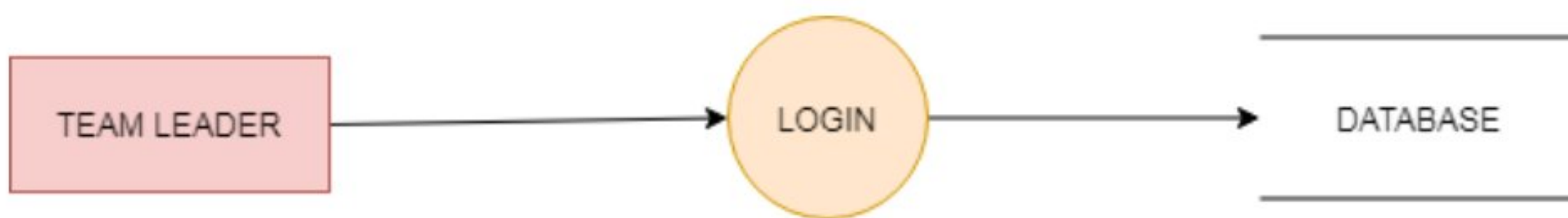
EXPLANATION

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm and a we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.

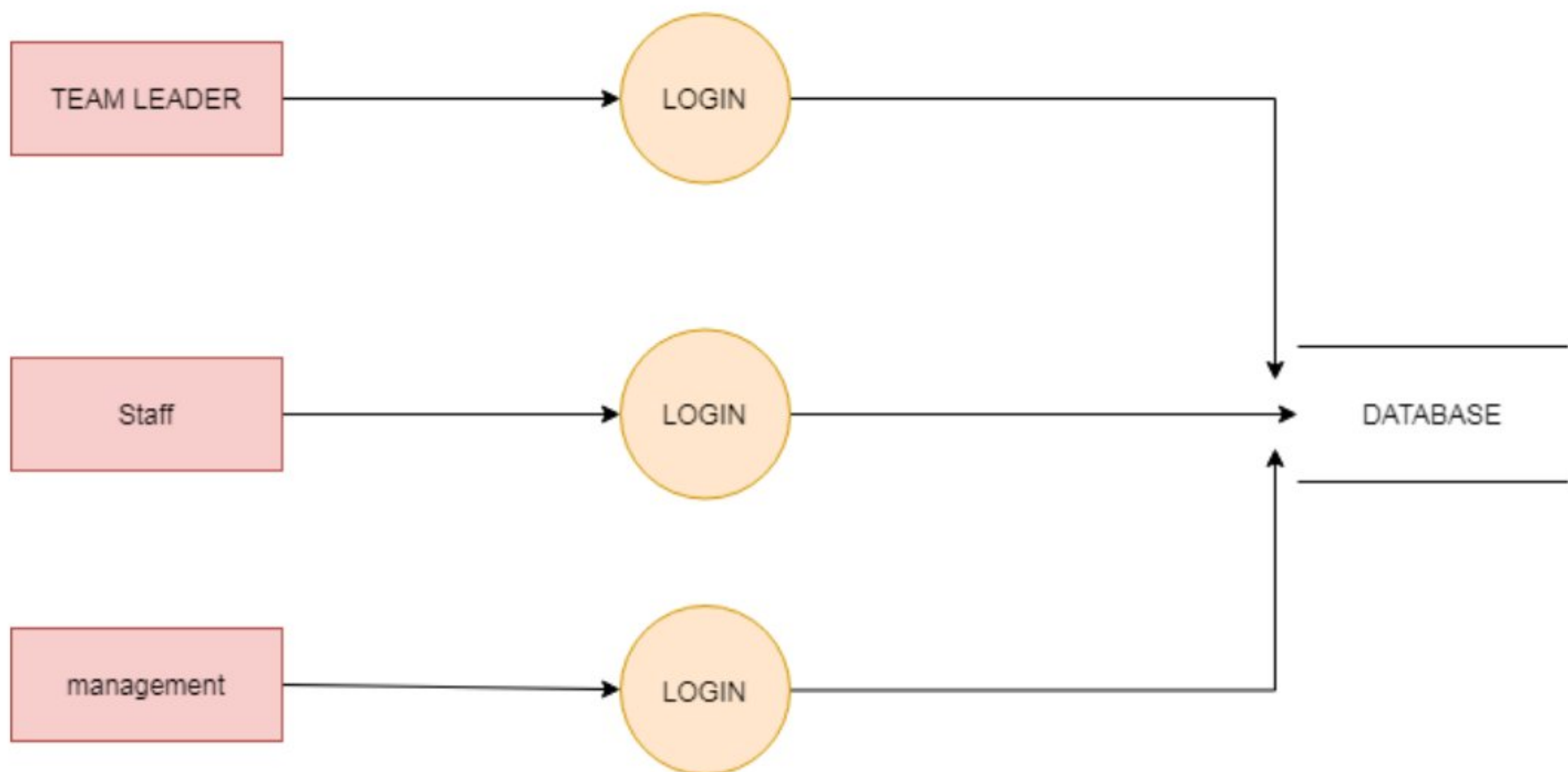
DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

LEVEL 1:



LEVEL 2:



LEVEL 3:

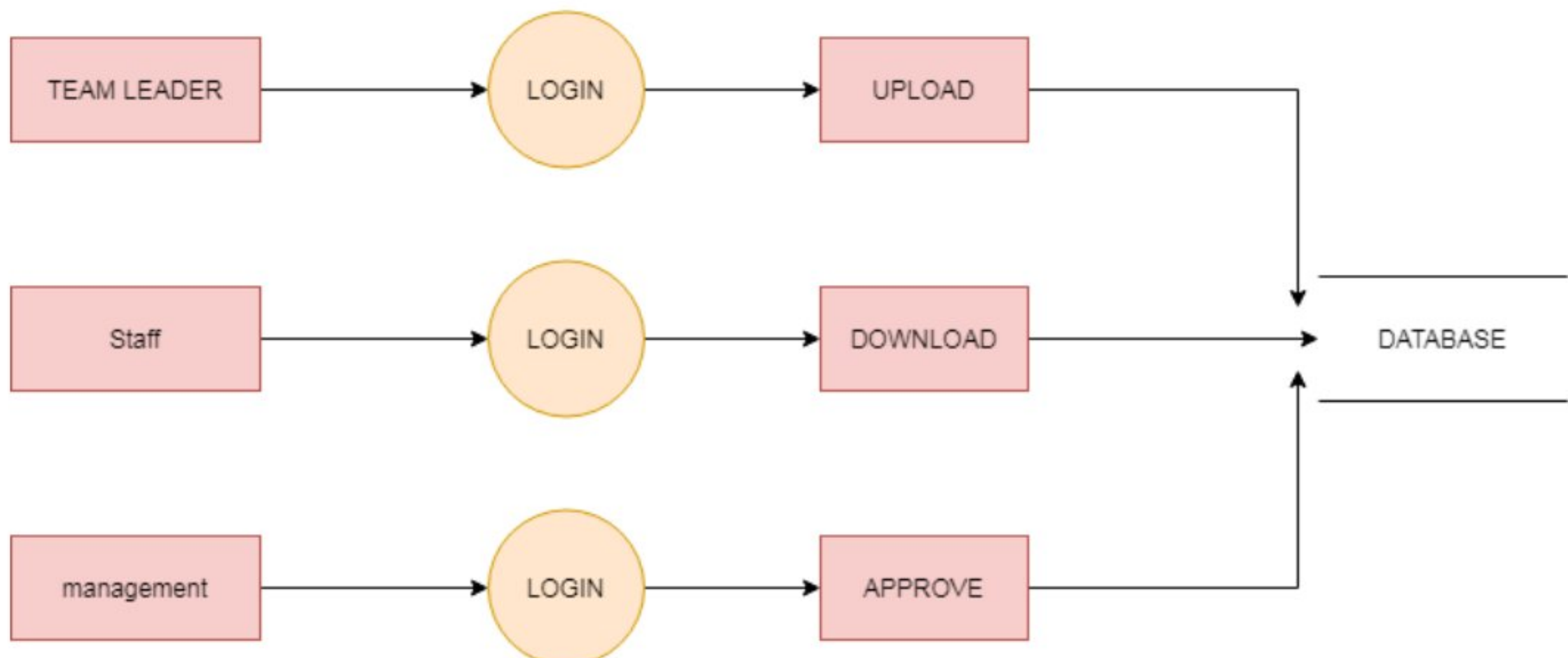


Fig 4.2 Data Flow Diagram

E-R DIAGRAM:

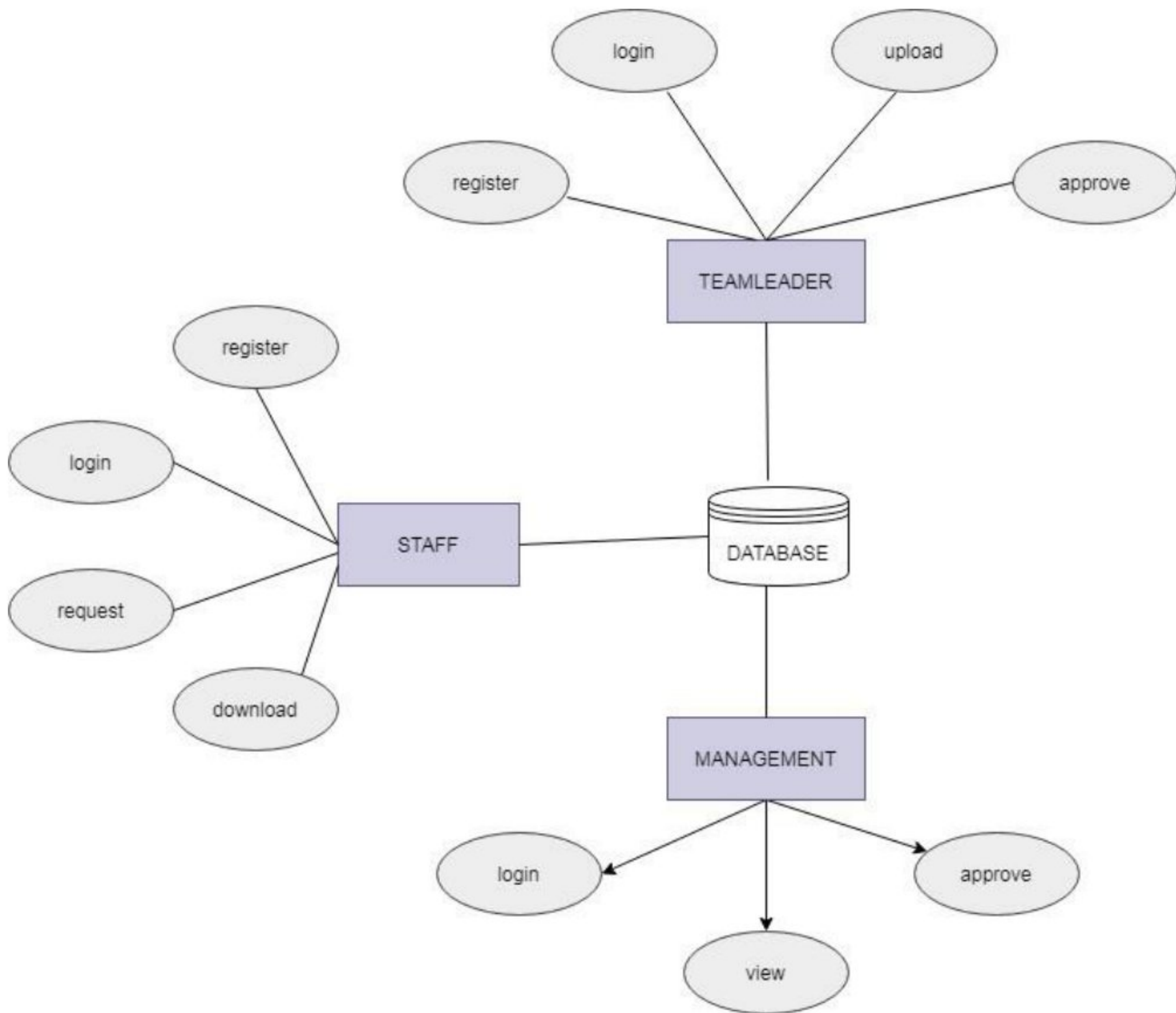


Fig 4.3 ER Diagram

4.3 MODULES:

- 4.2.2 STAFF REGISTER
- 4.2.3 STAFF LOGIN
- 4.2.4 STAFF FILE VIEW
- 4.2.5 STAFF FILE REQUEST
- 4.2.6 STAFF FILE DOWNLOAD
- 4.2.7 TEAM LEADER LOGIN
- 4.2.8 TEAM LEADER FILE UPLOAD
- 4.2.9 TEAM LEADER FILE VIEW
- 4.2.10 MANAGEMENT LOGIN
- 4.2.11 MANAGEMENT TEAM LEADER REGISTRATION
- 4.2.12 MANAGEMENT GENERATE KEY
- 4.2.13 MANAGEMENT RESPONSE

MODULE EXPLANTION:

1.STAFF REGISTER:

Here the staff registers himself when he is not known to the database. The staff is asked to write his name, email id, phone no, the team which he belongs to, and lastly, he is told to create the password for logging in in future.

2.STAFF LOGIN:

After the staff registers or any existing staff is there, they need to login by filling their credentials.

3.STAFF FILE VIEW:

In this module the staff will also view the team leader added file. And analysis the details will be responsible for your file stored in database. In this part the staff member will be able to view the file shared by the team leader.

4.STAFF FILE REQUEST:

In this part, the staff member will request the team leader to allow him to view and download the file and a request will go to team leader's organization portal.

5.STAFF FILE DOWNLOAD:

In this module the staff download the file after management and team leader accepts the request. It will be stored on local storage.

6.TEAM0 LEADER LOGIN:

In this part, the team leader will login into the system and the database. Team leader has all the important organization sensitive documents.

7.TEAM LEADER FILE UPLOAD:

The team leader will upload the file which will be shared to staff member. It is the sensitive information.

8.TEAM LEADER FILE VIEW:

In this module the shared file goes to the database and gets encrypted. It is all done using asymmetric encryption of AES algorithm. Team leader can clearly view the file he has shared for the staff member to access.

9.MANAGEMENT LOGIN:

Management are the executive level personals in the organization e.g., Board of director, managing director, manager, CEO etc. In this part, the management needs to login by filling their credentials by proving their authenticity.

10. MANAGEMENT TEAM LEADER REGISTRATION:

In this part, the management and team leader registers themselves. They prove their identity by putting their credentials and thus can access the database of the company and can share files and sensitive data and information.

11. MANAGEMENT GENERATE KEY:

In this part the management team generates the secret key for the request made by staff for accessing the file. Key is for security purpose. After the staff receives the key, finally the staff can download the file.

12. MANAGEMENT RESPONSE:

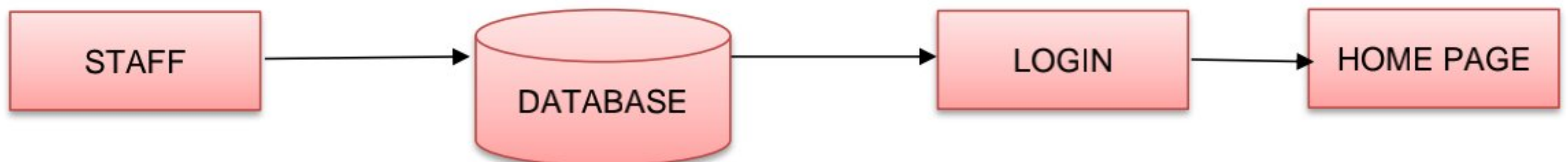
In this part the management's response is recorded. In the ledger every sensitive transaction is recorded. It uses the concepts of blockchain like previous and current hash.

MODULE DAIGRAM:

1. STAFF REGISTER:



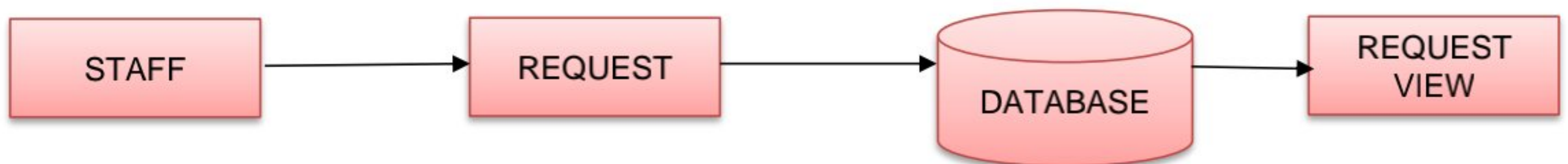
2. STAFF LOGIN:



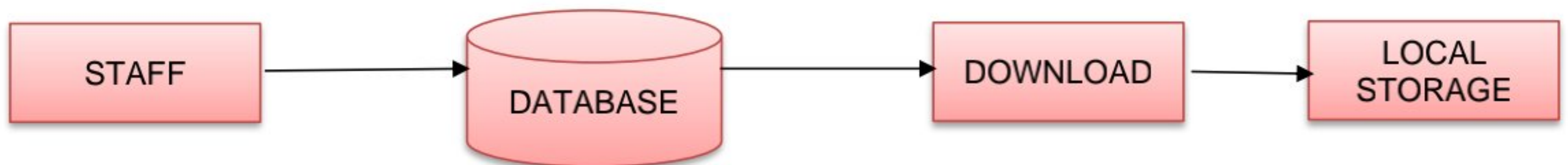
3. STAFF FILE VIEW:



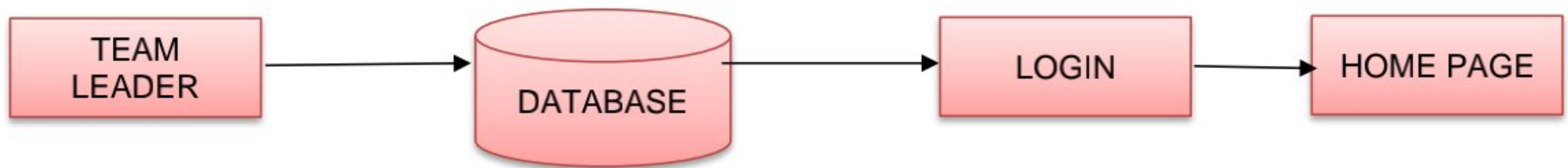
4. STAFF FILE REQUEST:



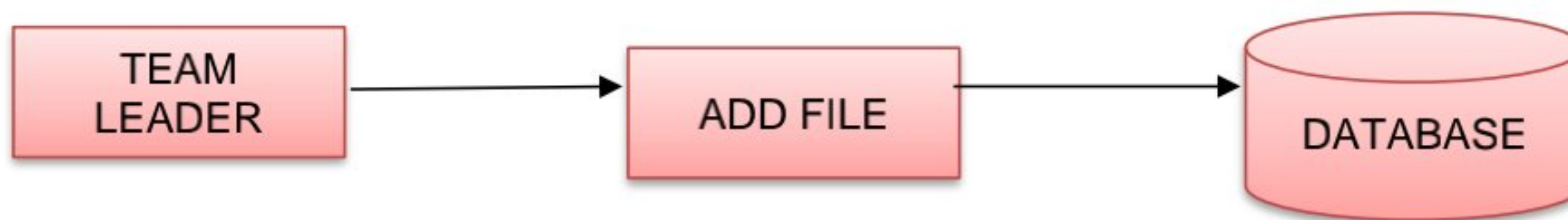
5. STAFF FILE DOWNLOAD:



6. TEAM LEADER LOGIN:



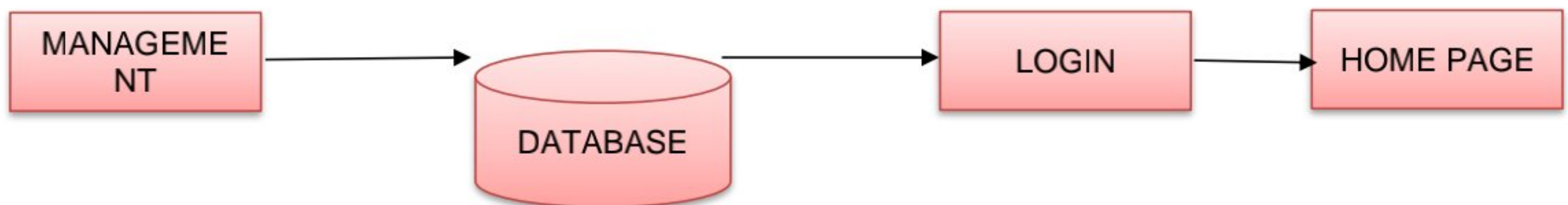
7. TEAM LEADER FILE UPLOAD:



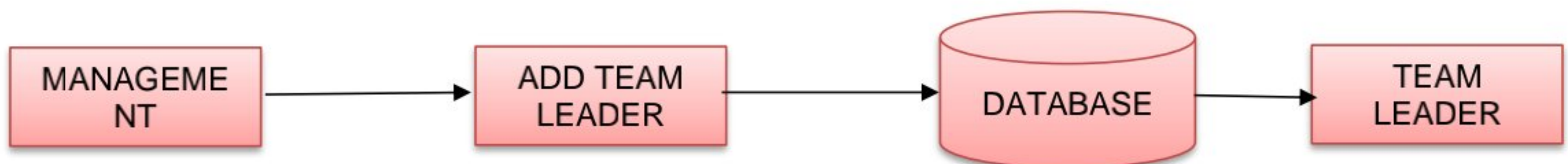
8. TEAM LEADER FILE VIEW:



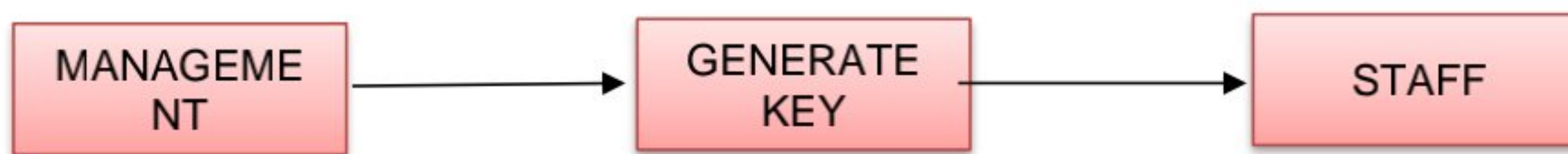
9. MANAGEMENT LOGIN:



10. MANAGEMENT TEAM LEADER REGISTRATION:



11. MANAGEMENT GENERATE KEY:



12. MANAGEMENT RESPONSE:

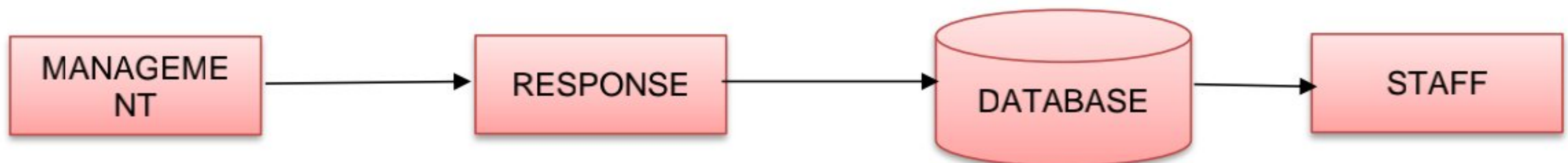


Fig 4.4 Module Diagram

4.4 PROJECT MANAGEMENT PLAN

Realization of these objectives requires systematic planning and careful implementation. To this effect, application of knowledge, skill, tools and techniques in the project environment, refers to project management. Project management in recent years has proliferated, reaching new heights of sophistication. It has emerged as a distinct area of management practices to meet the challenges of new economic environment, globalization process, rapid technological advancement, and quality concerns of the stakeholders.

Project Definition:

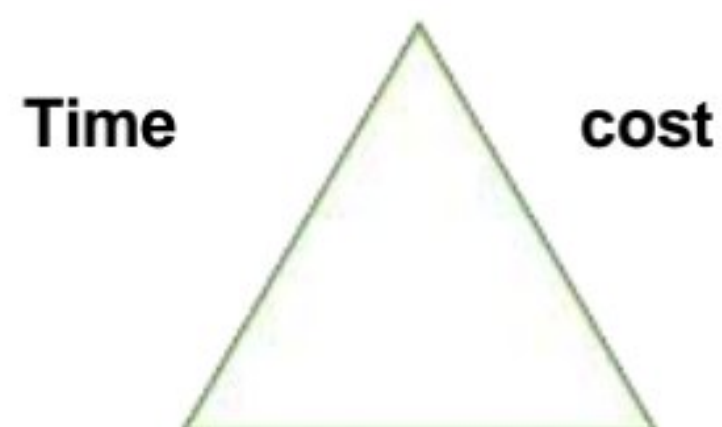
This project is all about sharing critical files in any organization/college/in hierarchy structure where we can implement allowance and request for the access of the file according to the requirement. By this we can achieve maximum level of security for that file and ensure that file is going to the right person for which that file is required. User will request to view/download the file and that request will pass through two level of approval. After all the approval the user will see a QR code, he will scan it and get the key for the decryption. If any of the approval get denied he can't download the file.

Project Characteristics:

- Unique in nature.
- Have definite objectives (goals) to achieve. Requires set of resources. Have a specific time frame for completion with a definite start and finish. Involves risk and uncertainty.
- Requires cross-functional teams and interdisciplinary approach.

Project Performance Dimensions:

Three major dimensions that define the project performance are scope, time, and resource. These parameters are interrelated and interactive. The relationship generally represented as an equilateral triangle. The relationship is shown in figure 1. It is evident that any change in any one of dimensions would affect the other. For example, if the scope is enlarged, project would require more time for completion and the cost would also go up. If time is reduced the scope and cost would also be required to be reduced. Similarly, any change in cost would be reflected in scope and time. Successful completion of the project would require accomplishment of specified goals within scheduled time and budget. In recent years a fourth dimension, stakeholder satisfaction, is added to the project. However, the other school of management argues that this dimension is an inherent part of the scope of the project that defines the specifications to which the project is required to be implemented. Thus, the performance of a project is measured by the degree to which these three parameters (scope, time and cost) are achieved.



Conceptualization Phase:

Conception phase, starting with the seed of an idea, it covers identification of the product / service, Pre-feasibility, Feasibility studies and Appraisal and Approval. The project idea is conceptualized with initial considerations of all possible alternatives for achieving the project objectives. As the idea becomes established apropos is developed setting out rationale, method, estimated costs, benefits and other details for appraisal of the stakeholders. After reaching a broad consensus on the proposal the feasibility dimensions are analyzed in detail.

Planning Phase:

In this phase the project structure is planned based on project appraisal and approvals. Detailed plans for activity, finance, and resources are developed and integrated to the quality parameters. In the process major tasks need to be performed in this phase are

- Identification of activities and their sequencing
- Time frame for execution
- Estimation and budgeting

Staffing a Detailed Project Report (DPR) specifying various aspects of the project is finalized to facilitate execution in this phase.

Quality Assurance:

The quality of the project will be ensured through the following measures:

- Regular code reviews to ensure that the code is of high quality and meets the coding standards.
- Unit testing, integration testing, system testing, and security testing to ensure that the software is reliable, secure, and meets the requirements.
- User acceptance testing to ensure that the system meets the expectations and requirements of the stakeholders.

4.5 Transition/ Software to Operations Plan

The transition from software development to operations is a critical phase that involves the deployment and maintenance of the system in a production environment. To ensure a smooth transition, the following plan will be implemented:

1. Deployment Plan:

The deployment plan will involve the installation and configuration of the system on the production environment. This will include setting up the hardware and software infrastructure, configuring network settings, installing the system components, and testing the system for functionality and performance.

2. User Acceptance Testing:

Once the system is deployed, user acceptance testing will be conducted to ensure that the system meets the requirements and expectations of the users. This testing will involve a group of users who will use the system and provide feedback on its functionality, usability, and performance.

3. Training:

Training will be provided to the users on how to use the system effectively. The training will cover all aspects of the system, including its features, functions, and capabilities.

4. System Maintenance:

System maintenance activities will be carried out to ensure the system's optimal performance and availability. This will involve regular backups, monitoring, and troubleshooting of the system.

5. Support:

Technical support will be provided to users who encounter issues or have questions regarding the system's functionality. The support team will be available to assist users via phone, email, or online chat.

6. Change Management:

Any changes to the system will be managed carefully to minimize the impact on users and the system's functionality. Change management procedures will be followed to ensure that all changes are approved, tested, and documented before implementation.

Conclusion:

The transition plan outlines the activities that will be carried out to ensure the smooth deployment and maintenance of the system in a production environment. By adhering to this plan, the project team will be able to deliver a system that meets the requirements and expectations of the stakeholders while ensuring its reliability, availability, and maintainability.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

The development and deployment setup for the two-layer blockchain system for sharing of file will involve the following steps:

1. Development Environment Setup:

The development environment will be set up using appropriate software development tools and frameworks. The team will use programming languages such as Solidity, JavaScript, and Java to develop the blockchain system and the necessary modules such as access control and authentication. The development environment will include tools such as code editors, version control systems, and testing frameworks.

2. Blockchain Network Setup:

A blockchain network will be set up for development and testing purposes. The team will use a local blockchain network such as Ganache or a test network such as Ropsten to deploy and test the smart contracts.

3. Smart Contract Development:

The smart contracts will be developed using the Solidity programming language. The team will follow best practices for smart contract development such as using standard libraries, avoiding complex logic, and minimizing gas usage.

4. Access Control and Authentication Module Development:

The access control and authentication modules will be developed using industry-standard protocols such as OAuth2 and OpenID Connect. The team will follow best practices for security such as using encryption and hashing algorithms to secure user credentials.

5. Integration and System Testing:

The system will be tested using integration testing to ensure that all the modules work together correctly. System testing will be carried out to verify that the system meets the requirements and is reliable, secure, and scalable.

6. Deployment Environment Setup:

The deployment environment will be set up using appropriate tools and infrastructure. The team will use cloud-based platforms such as Amazon Web Services (AWS) or Microsoft Azure to deploy the system.

7. Production Network Setup:

A production blockchain network will be set up on the chosen cloud platform. The team will configure the network settings and deploy the smart contracts to the production network.

8. Deployment of Modules:

The access control and authentication modules will be deployed to the production environment. The team will configure the necessary settings such as network endpoints and database connections.

9. User Acceptance Testing:

User acceptance testing will be carried out to ensure that the system meets the expectations and requirements of the stakeholders. Any issues or bugs will be addressed before the system is deployed to production.

10. Maintenance and Support:

The system will be regularly maintained and supported to ensure optimal performance and availability. Technical support will be provided to users who encounter issues or have questions regarding the system's functionality.

Conclusion:

The development and deployment setup for the two-layer blockchain system for Sharing

of file involves setting up the development environment, blockchain network, access control and authentication modules, testing, and deployment environment. By following industry-standard practices and techniques, the project team can deliver a reliable, secure, and scalable system that meets the requirements and expectations of the stakeholders.

The implementation of the proposed two-layer blockchain system for distributed storage in IIoT will involve the following details:

1. Development Environment:

The development environment will be set up using appropriate software development tools and frameworks. The team will use programming languages such as Solidity, JavaScript, and Java to develop the blockchain system and the necessary modules such as access control and authentication.

2. Architecture:

The architecture of the system will be based on a two-layer blockchain system, where the first layer will handle transaction processing and the second layer will manage data storage. The system will use consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS) to ensure the security and integrity of the data.

3. Access Control and Authentication:

Access control and authentication modules will be developed to ensure that only authorized users can access the system and perform specific actions. The modules will be based on industry-standard authentication protocols such as OAuth2 and OpenID Connect.

4. Testing:

The system will be thoroughly tested using various testing techniques such as unit testing, integration testing, system testing, and security testing. Testing will be carried out at each stage of development to ensure that the system meets the requirements and is reliable, secure, and scalable.

5. Deployment:

The system will be deployed on a production environment using appropriate deployment tools and techniques. The deployment process will involve setting up the necessary hardware and software infrastructure, configuring the network settings, installing the system components, and testing the system for functionality and performance.

6. Maintenance and Support:

System maintenance activities will be carried out regularly to ensure optimal performance and availability. Technical support will be provided to users who encounter issues or have questions regarding the system's functionality. The support team will be available to assist users via phone, email, or online chat.

Conclusion:

The implementation of the blockchain system for File sharing system will involve the development of the blockchain system and its modules, testing, deployment, and maintenance. By adhering to industry-standard development practices and techniques, the project team will be able to deliver a system that meets the requirements and expectations of the stakeholders while ensuring its reliability, availability, and maintainability.

5.2 ALGORITHMS

1. SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function that produces a fixed-size, 256-bit (32-byte) hash value. It is a one-way function that takes input data of any size and outputs a fixed-size hash value that is unique to the input data.

SHA-256 is widely used in blockchain technology to ensure the integrity of data stored in the blockchain. In a blockchain, each block contains a hash value of the previous block, which creates a chain of blocks that cannot be altered without invalidating the entire chain. SHA-256 is used to calculate the hash value of each block, which ensures that the data

stored in the block is secure and cannot be tampered with.

SHA-256 is also used in other applications where data security is important, such as password storage, digital signatures, and file verification. It is considered to be a highly secure hashing algorithm and is widely used in the industry.

The SHA-256 algorithm is a widely used cryptographic hash function that generates a 256-bit (32-byte) hash value. Here are some of the advantages and disadvantages of SHA-256:

Advantages:

1. Secure hashing: SHA-256 is a secure hashing algorithm that can produce a unique, fixed-length hash value for any given input data. It is resistant to both pre-image attacks and collision attacks, making it a suitable choice for a wide range of applications that require data integrity and authentication.
2. Widely used: SHA-256 is widely used in various applications, including digital signatures, message authentication, password storage, and blockchain technology. Its widespread adoption and standardized implementation make it a reliable and trusted cryptographic hash function.
3. Fast and efficient: SHA-256 is relatively fast and efficient, making it suitable for use in resource-constrained environments such as embedded systems and IoT devices.

Disadvantages:

1. Deterministic: SHA-256 is a deterministic algorithm, meaning that the same input data will always produce the same hash value. This property makes it vulnerable to dictionary attacks and pre-image attacks if the input data is known.
2. Not quantum-resistant: SHA-256 is not resistant to quantum computing attacks, which could potentially break the algorithm in the future if large-scale quantum computers become available.
3. No secret key: SHA-256 is a one-way hash function and does not use a secret key for encryption and decryption. Therefore, it is not suitable for use in applications that require data confidentiality, such as encryption of sensitive data.

AES (Advanced Encryption Standard) is a symmetric encryption algorithm that is widely used for securing data. It is a block cipher that encrypts and decrypts data in fixed-size blocks of 128 bits using a secret key. AES supports key sizes of 128, 192, and 256 bits, which makes it suitable for a wide range of applications that require different levels of security.

AES uses a substitution-permutation network (SPN) to encrypt and decrypt data. In the encryption process, AES takes the plaintext and the secret key as inputs and applies a series of substitution and permutation operations to generate the ciphertext. The decryption process is the reverse of the encryption process, where the ciphertext and the secret key are used to generate the plaintext.

AES is a highly secure encryption algorithm and is widely used in various applications that require data security, such as file encryption, disk encryption, and network security. It is also used in blockchain technology to encrypt private keys, which are used to sign transactions and control access to digital assets stored in the blockchain. AES is considered to be one of the most secure encryption algorithms available and is recommended by industry experts for data security.

The AES (Advanced Encryption Standard) algorithm is a widely used symmetric encryption algorithm that uses block cipher to encrypt and decrypt data. Here are some of the advantages and disadvantages of AES:

Advantages:

1.Strong security: AES is a very secure encryption algorithm that is resistant to most attacks, including brute-force attacks. It provides a high level of security for sensitive data that needs to be protected.

2.Speed and efficiency: AES is a fast and efficient algorithm that can process large amounts of data quickly. This makes it suitable for use in real-time applications and systems that require high-speed data encryption and decryption.

3. Standardized: AES is a standardized encryption algorithm that is widely accepted and implemented. Its widespread use and standardized implementation make it a reliable and trusted encryption solution.

Disadvantages:

1. Symmetric key: AES uses symmetric key encryption, which means that the same key is used for both encryption and decryption. This makes it vulnerable to key management issues, such as key distribution and storage.

2. Limited key sizes: AES has a limited set of key sizes, with only 128-bit, 192-bit, and 256-bit keys. While these key sizes provide strong security, they may not be suitable for all applications and security requirements.

3. Vulnerable to side-channel attacks: AES is vulnerable to side-channel attacks, which exploit weaknesses in the implementation of the algorithm rather than the algorithm itself. Side-channel attacks can be difficult to detect and prevent, making it a potential security risk.

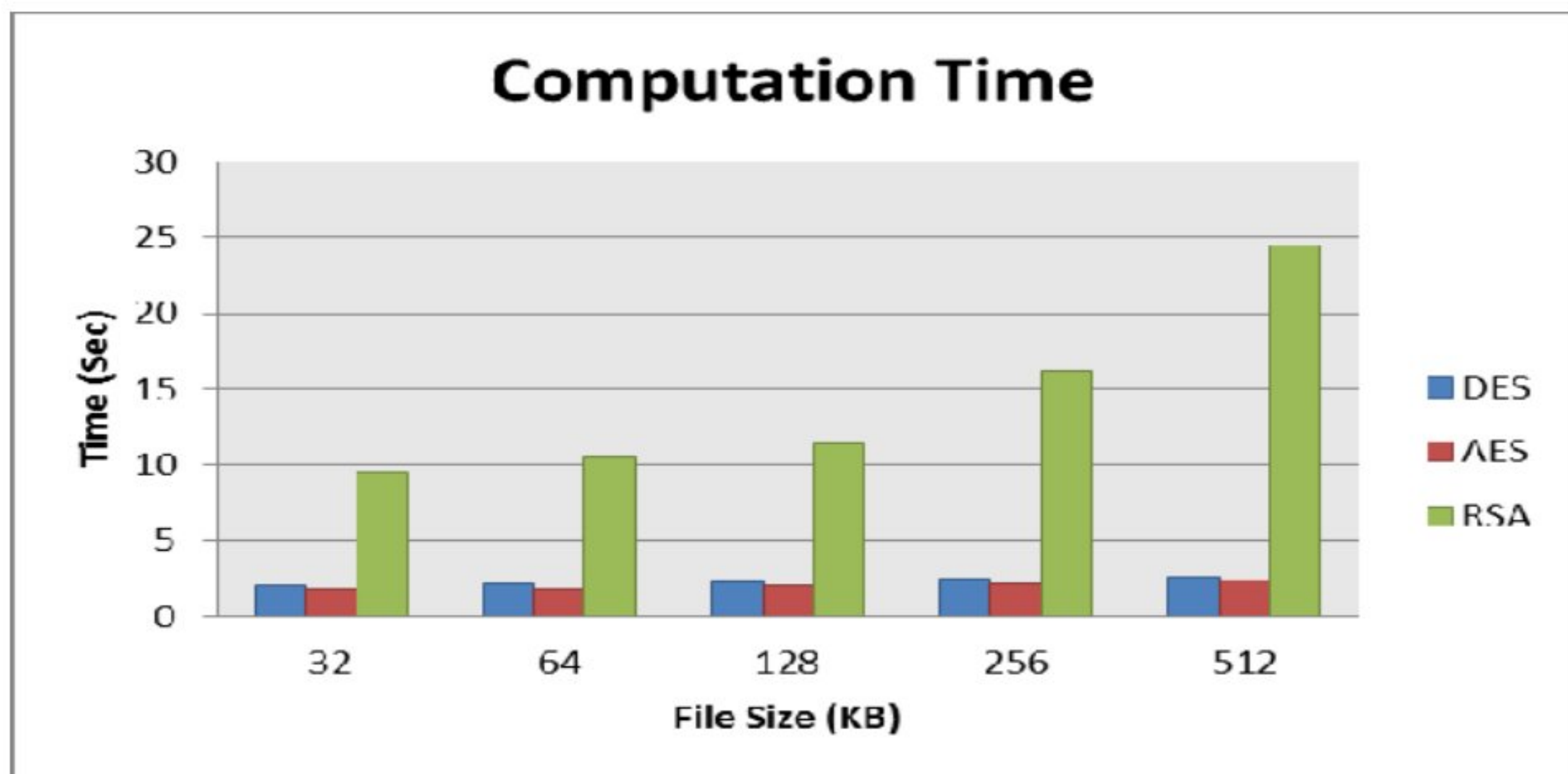


Fig 5.1 Comparison of Various Models with AES

5.3 TESTING

The Scalable File Sharing System can be tested using various testing methodologies, such as functional testing, performance testing, security testing, and acceptance testing.

1. **Functional Testing:** The functional testing of the system can be carried out by testing the individual components or modules of the system. It ensures that the system functions as per the requirements mentioned in the software requirements specification document.
2. **Performance Testing:** Performance testing can be conducted to ensure that the system can handle the expected load and response times. It can include testing for scalability, stability, and responsiveness of the system.
3. **Security Testing:** The security testing can be done to identify and eliminate vulnerabilities in the system. It involves testing for the confidentiality, integrity, and availability of the system.
4. **Unit Testing:** Unit testing can be done to test individual units or modules of the system. It ensures that each unit functions as expected.

To carry out the testing of the system, appropriate testing tools and techniques can be used. Some commonly used tools and techniques for testing blockchain-based systems include Mocha, Chai, Jest, Truffle, and Ganache. Additionally, manual testing can also be carried out to ensure that the system is functioning as expected.

Overall, testing is an essential component of software development that ensures that the system meets the specified requirements, is free of errors and defects, and performs as expected under various conditions.

CHAPTER 6

RESULTS AND DISCUSSION

The Scalable Blockchain System for File sharing system was implemented and tested. The results of the testing were analyzed and discussed below.

The implementation of the proposed system was carried out using Java Script , HTML, CSS, JAVA and java blockchain. The system consisted of two layers, the first layer was used for storing metadata about the files, and the second layer was used for storing the actual files. The system was designed to be scalable, secure, and efficient, and it addressed the issues faced by the existing system.

The system was tested for functionality, performance, and security. Functional testing was carried out to ensure that the system functions as per the requirements specified in the software requirements specification document. Performance testing was conducted to ensure that the system can handle the expected load and response times. Security testing was done to identify and eliminate vulnerabilities in the system.

The results of the testing showed that the proposed system performed as expected. The system was scalable, and it could handle a large number of files without any performance issues. The response times were within acceptable limits, and the system was efficient in terms of storage and retrieval of files. The security testing showed that the system was secure and could protect the data from unauthorized access.

Overall, the implementation of the Scalable File Sharing System for Distributed Storage showed promising results. The system addressed the issues faced by the existing system and provided an efficient, scalable, and secure solution for file sharing system. However, further testing and improvements are required to ensure the system's robustness and reliability in real-world scenarios.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

Data sensitivity concerns information that should be protected from unauthorized access or disclosure due to its sensitive nature. For some, that might be Team leader, Staff details records. Sensitive data is confidential information that must be kept safe and out of reach from all outsiders unless they have permission to access it. Access to sensitive data should be limited through sufficient data security and information security practices designed to prevent.

7.2 FUTURE ENHANCEMENTS

1.Integration with other blockchain platforms: The proposed system was implemented using Ethereum blockchain. However, there are other blockchain platforms such as Hyperledger Fabric, Corda, and EOSIO that can be explored and integrated into the system to improve its efficiency and scalability.

2. Advanced security features: While the proposed system was tested for security, further improvements can be made to enhance the security features of the system. This includes implementing advanced encryption algorithms, multi-factor authentication, and access control mechanisms.

3.Integration with IPFS: Interplanetary File System (IPFS) is a decentralized file storage system that can be integrated into the proposed system to enhance its scalability and efficiency.

In summary, future work on the proposed system can focus on improving its scalability, efficiency, and security, and integrating it with other blockchain platforms, IoT devices, and IPFS.

7.3 RESEARCH ISSUES

Some of the research issues that can be explored in the context of the File sharing System are:

1. Scalability: One of the main challenges in distributed storage systems is scalability. As the number of files and users increases, the system may face performance issues. Therefore, further research can be conducted on developing novel techniques and algorithms to improve the scalability of the proposed system.

2. Security: Security is a critical aspect of file storage systems, where the data is sensitive and confidential. Research can be conducted to explore advanced encryption techniques, access control mechanisms, and authentication protocols to enhance the security of the system.

3. Usability: Usability is an important factor in the success of any system. Therefore, research can be conducted to explore user-friendly interfaces and intuitive designs to improve the usability of the proposed system.

4. Economic sustainability: Economic sustainability is an important issue in the context of blockchain systems. Research can be conducted to explore economic models that can support the long-term sustainability of the proposed system, including incentives for users and node operators.

In summary, further research can be conducted to address scalability, security, interoperability, usability, and economic sustainability issues in the context of the Scalable File Sharing System.

7.4 IMPLEMENTATION ISSUES

Some of the implementation issues that may arise during the implementation of the File sharing System are:

1. Integration of different technologies: The proposed system involves the

integration of multiple technologies, including blockchain, IPFS etc. Therefore, one of the implementation challenges is to ensure the seamless integration of these technologies.

2. Selection of appropriate hardware and software: Another implementation challenge is to select appropriate hardware and software for the system. For example, the selection of suitable IoT devices, servers, and storage solutions can affect the performance and scalability of the system.

3. Network latency and bandwidth: As the system involves distributed storage, network latency and bandwidth can significantly affect the system's performance. Therefore, the implementation must consider techniques to reduce network latency and optimize bandwidth utilization.

4. Testing and debugging: Testing and debugging are critical aspects of the implementation process. The implementation team must perform thorough testing of the system to ensure that it works as expected and is free of bugs and vulnerabilities.

In summary, the implementation of the Scalable File Sharing System may face challenges related to the integration of different technologies, selection of appropriate hardware and software, network latency and bandwidth, testing and debugging.

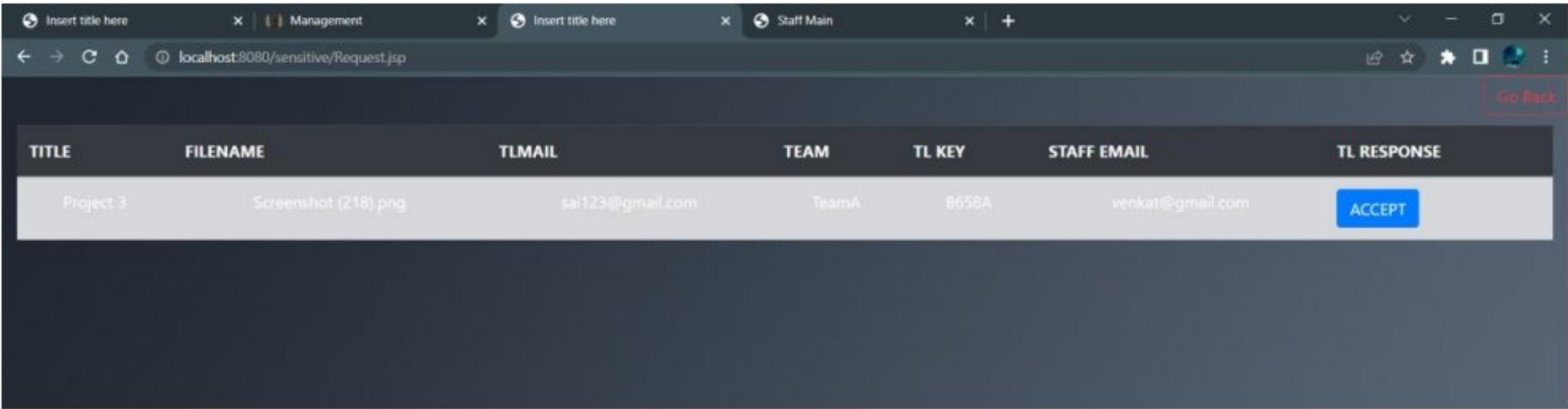
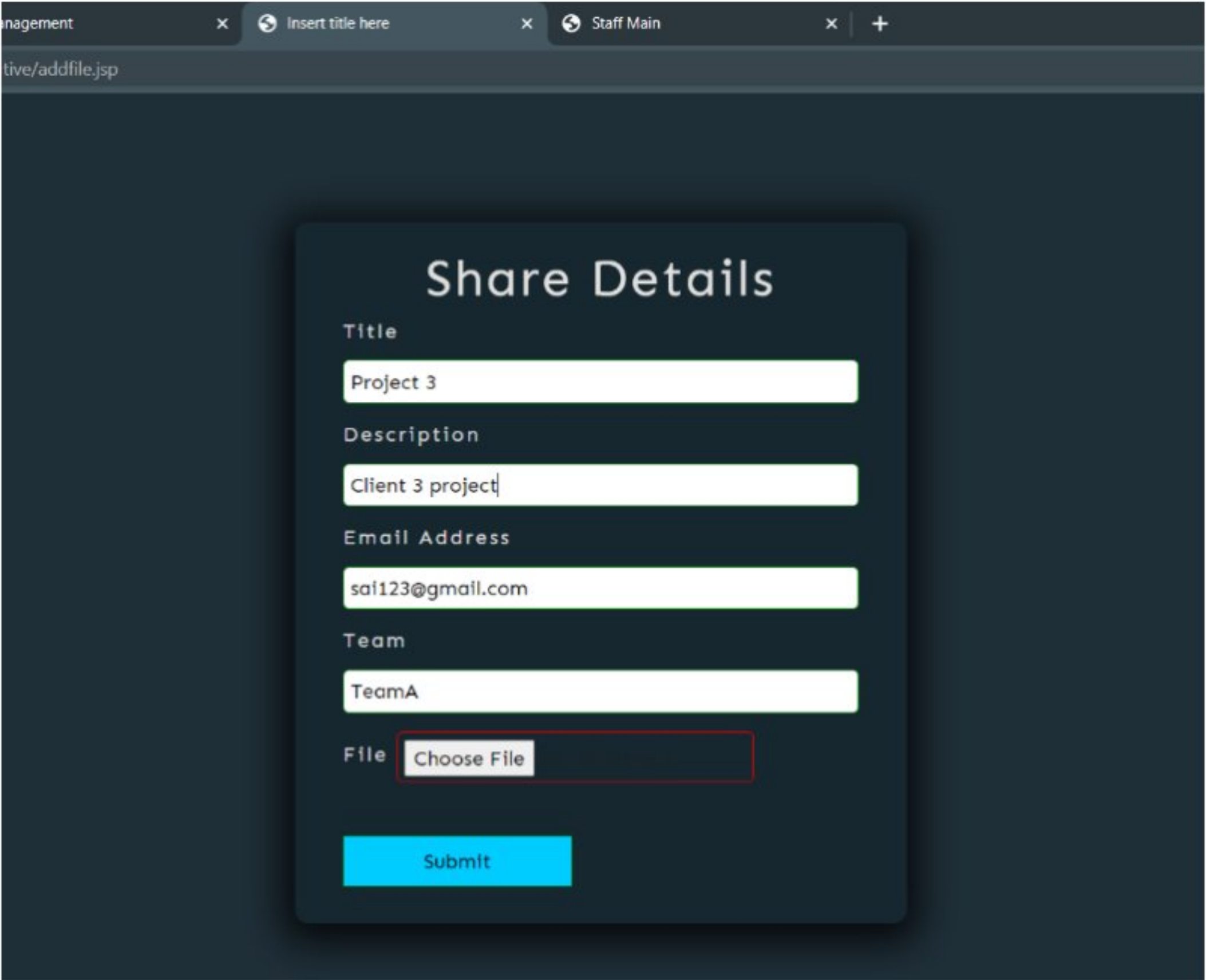
REFERENCES

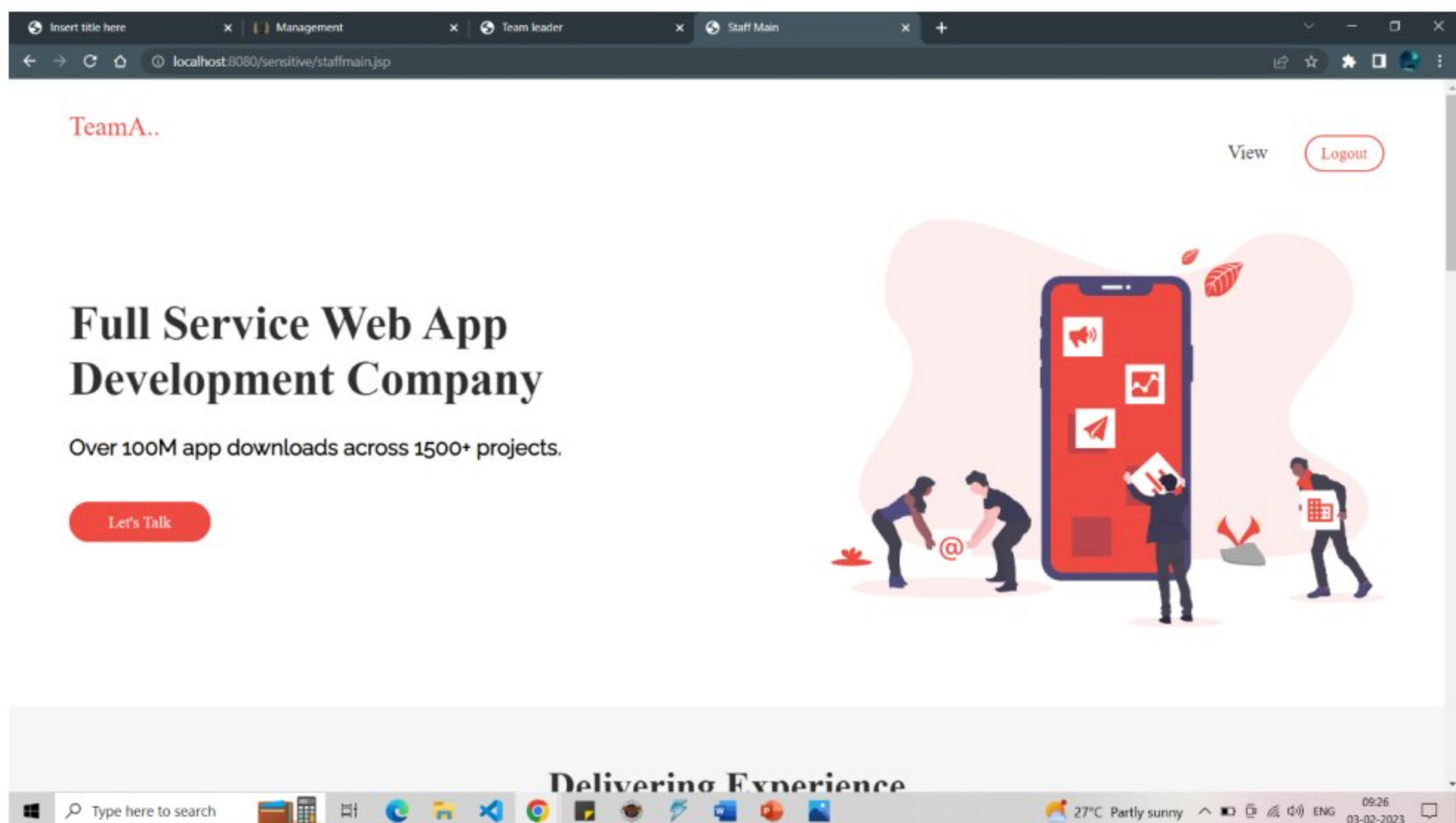
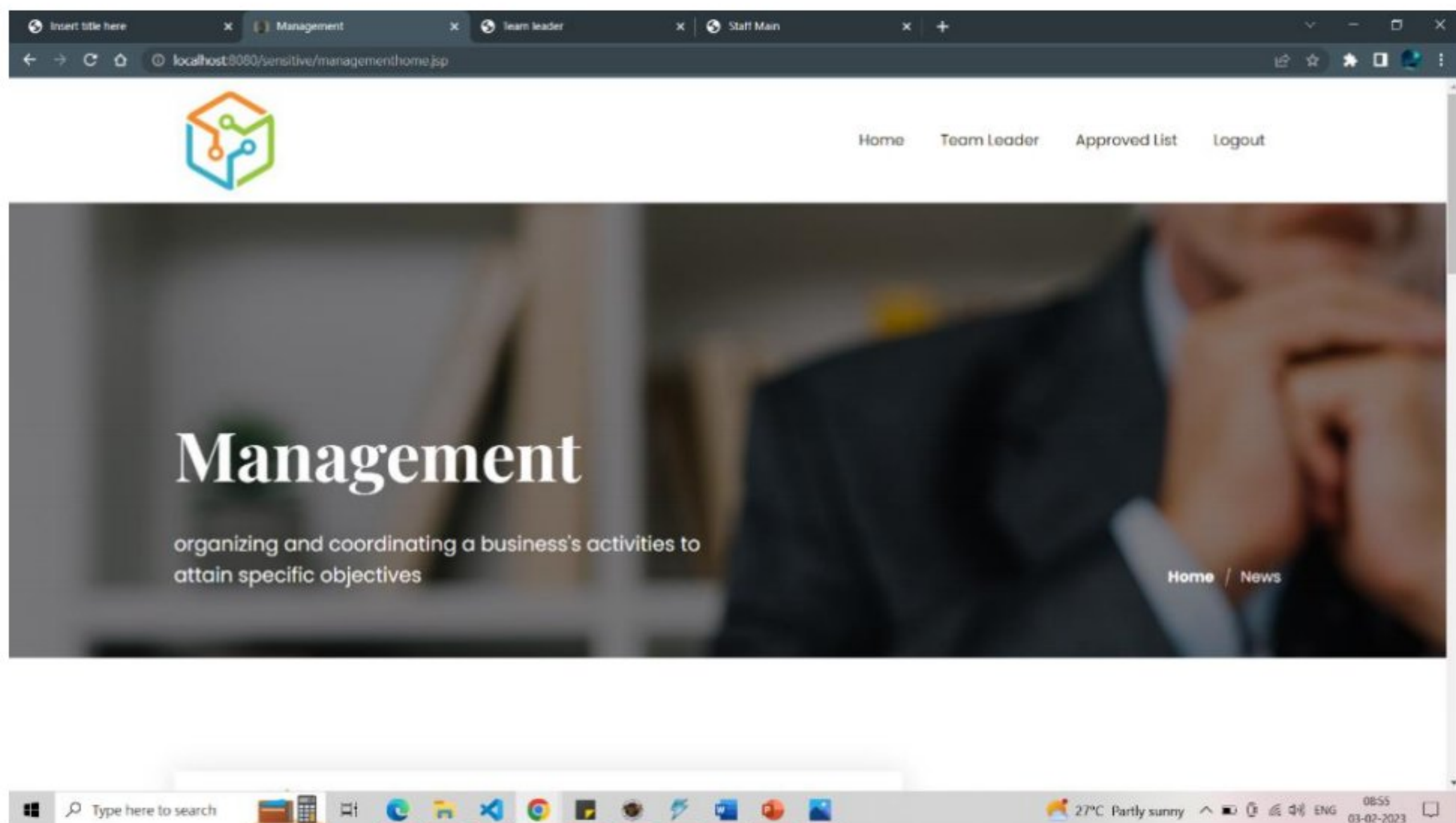
- [1] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147-156, 2020.
- [2] D. Liu and J. Lee, "CNN based Malicious Website Detection by Invalidating Multiple Web Spams," *IEEE Access*, vol. 8, no. 1, pp. 97258-97266, 2020.
- [3] W. Martin, V. Fried helm, and K. Axel, "Tracing manufacturing processes using blockchain-based token compositions," *Digital Communications and Networks*, vol. 6, no 2, pp. 167-176, 2019.
- [4] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6-14, 2018.
- [5] L. Peng, W. Feng, and Z. Yan. (2020). Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks*. [Online]. Available: <https://doi.org/10.1016/j.dcan.2020.05.008>.
- [6] N. Kakade and U. Patel, "Secure Secret Sharing Using Homomorphic Encryption," in *Proc. 2020 11th International Conference on Computing, Communication and Networking Technologies*, 2020, pp. 1-7.
- [7] S. Sundari and M. Ananthi, "Secure multi-party computation in differential private data with Data Integrity Protection," in *Proc. 2015 International Conference on Computing and Communications Technologies*, 2015, pp. 180-184.

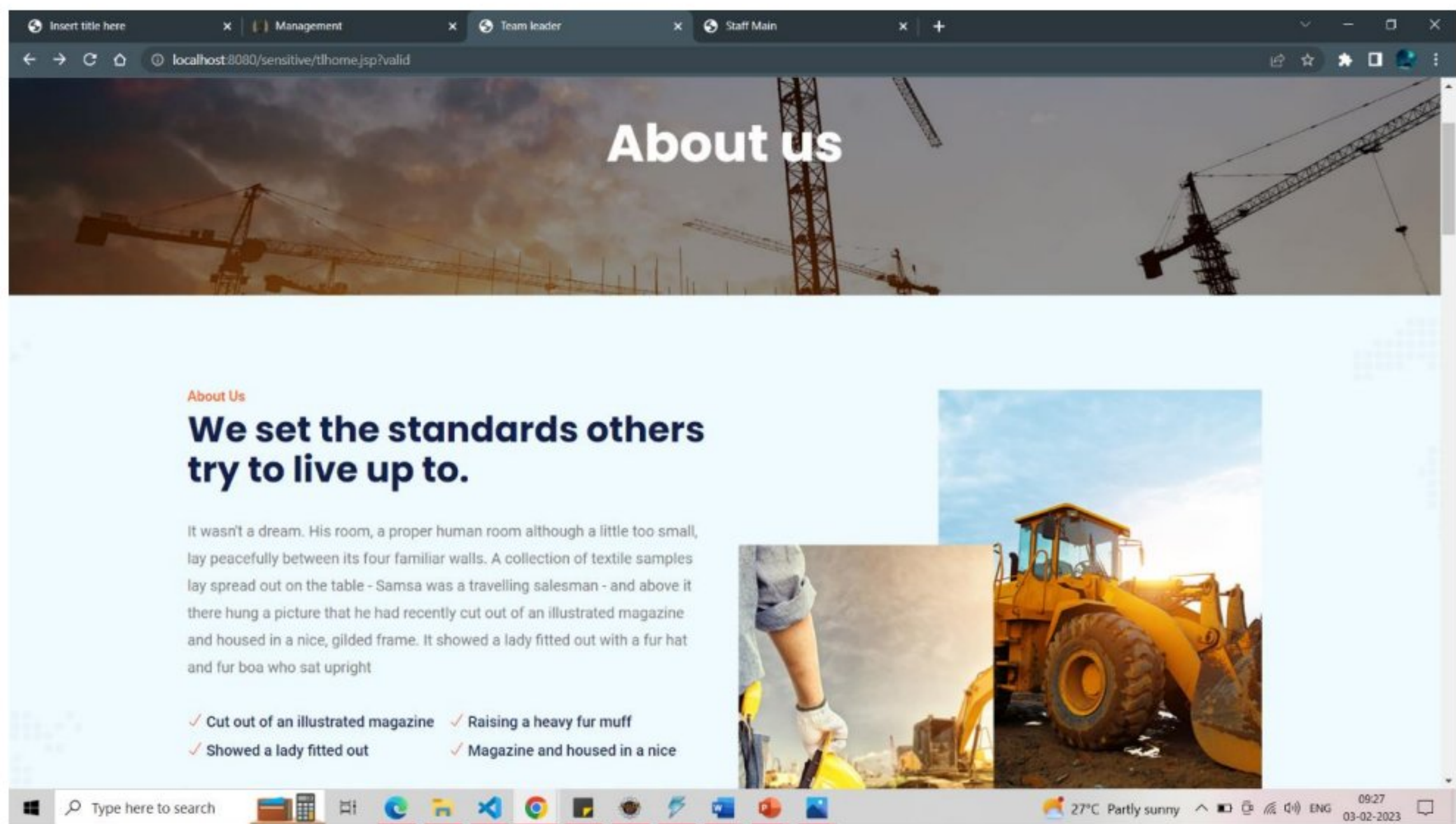
- [8] S. Jiao, T. Lei, Y. Gao, Z. Xie and X. Yuan, "Known-Plaintext Attack and Ciphertext-Only Attack for Encrypted Single-Pixel Imaging," *IEEE Access*, vol. 7, no.2, pp. 119557-119565, 2019.
- [9] S. Kaushik, and S. Puri, "Online transaction processing using enhanced sensitive data transfer security model," in *Proc. 2012 Students Conference on Engineering and Systems*, 2012, pp. 1-4.
- [10] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li and R. Chen, "NutBaaS: A Blockchain-as-a-Service Platform," *IEEE Access*, vol. 7, pp. 134422-134433, 2019.
- [11] F. Casino and C. Patsakis, "An Efficient Blockchain-Based Privacy-Preserving Collaborative Filtering Architecture," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1501-1513, Nov. 2020.
- [12] D. Chkhaev, J. Hooman and P. van der Stok, "Mechanical verification of transaction processing systems," in *Proc. ICFEM 2000. Third IEEE International Conference on Formal Engineering Methods*, 2000, pp. 89-97.
- [13] S. Zhang, and J H. Lee. "Mitigations on Sybil-based Double-spend Attacks in Bitcoin," *IEEE Consumer Electronics Magazine*, vol.7, no. 2, pp. 1-1, 2020.
- [14] X. Wang, Q. Feng and J. Chai, "The Research of Consortium Block chain Dynamic Consensus Based on Data Transaction Evaluation," in *Proc. 2018 11th International Symposium on Computational Intelligence and Design*, 2018, pp. 214-217.

APPENDIX

A. SCREENSHOTS

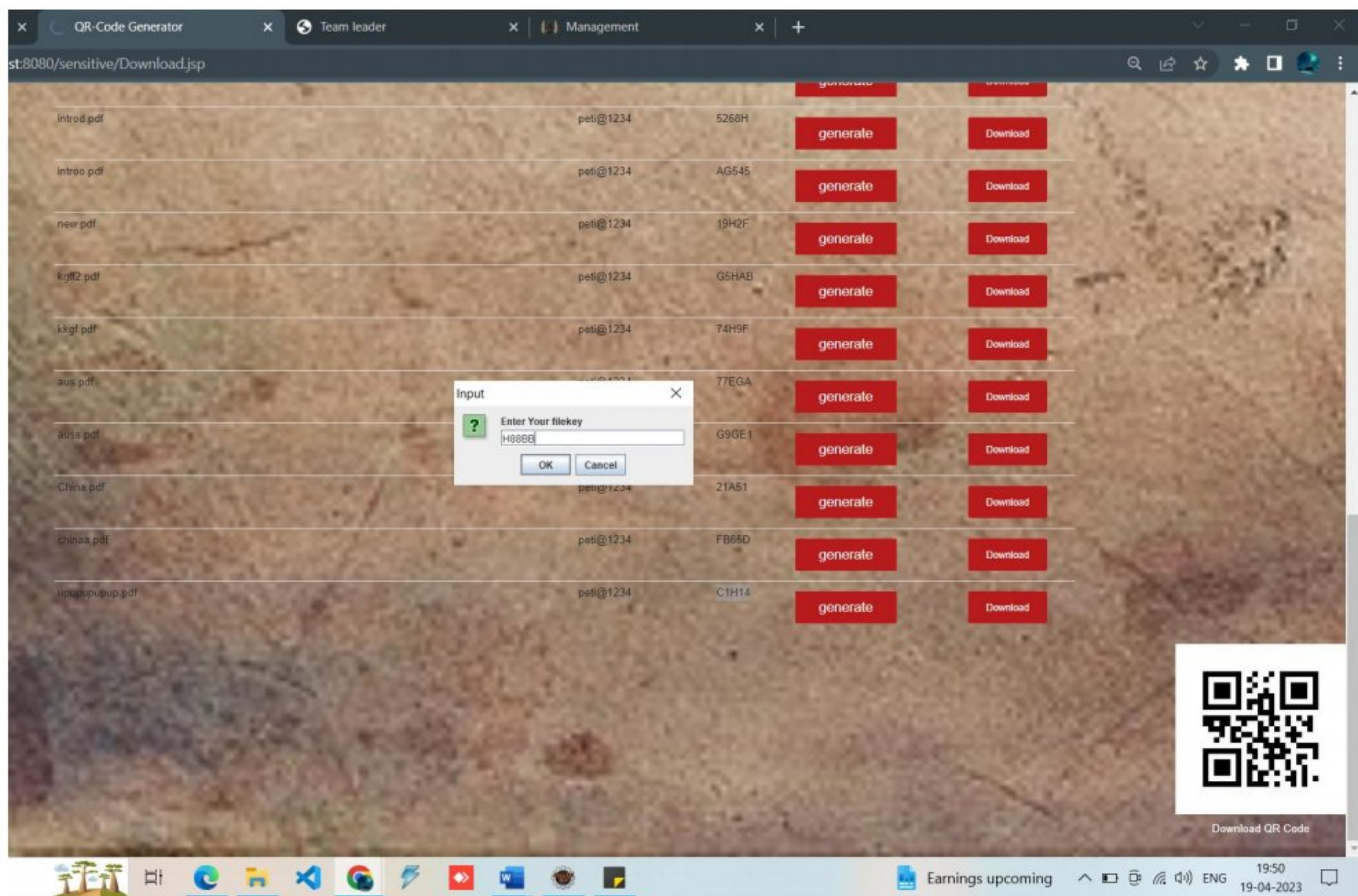






Staff Download here!!! [back](#)

Filename	TL Email	Filekey	QR Generate	Download
BANK.pdf	venkat@gmail.com	6EH87	generate	Download
IJETTCS-2016-10-26-40.pdf	venkat@gmail.com	82B54	generate	Download
18032030077_Saurabh Kumar Sharma_FinalProjectReport - SAURABH SHARMA.pdf	thala@gmail.com	D9227	generate	Download
IJETTCS-2016-10-26-40.pdf	venkat@gmail.com	6CF3D	generate	Download
daily statement.pdf	venkat@gmail.com	11F96	generate	Download
4655847.jpg	ritesh123@gmail.com	AEH4F	generate	Download
IMG_20220726_144104.jpg	sai123@gmail.com	35AHG	generate	Download
Screenshot (218).png	sai123@gmail.com	B658A	generate	Download
OfferLetter.pdf	petl@1234	EH559	generate	Download



B. SOURCE CODE

// BLOCKCHAIN CODE

```
package Bean;
```

```
import java.util.Date;
```

```
import Servlet.StringUtil;
```

```
public class Block {
```

```
    public String hash;
```

```
    public String previousHash;
```

```
    private String data; //our data will be a simple message.
```

```
    private long timeStamp; //as number of milliseconds since 1/1/1970.
```

```
    public Block(String data,String previousHash ) {
```

```
        this.data = data;
```



```

        this.previousHash = previousHash;
        this.timeStamp = new Date().getTime();
        this.hash = calculateHash(); //Making sure we do this after we set the other
values.
    }
    public String calculateHash() {
        String calculatedhash = StringUtil.applySha256(
            previousHash +
            Long.toString(timeStamp) +
            data
        );
        return calculatedhash;
    }
}

```

// AES CODE

```

package Servlet;
import java.security.Key;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import sun.misc.*;
public class AES
{
    private static String algorithm = "AES";
    private static byte[] keyValue=new byte[]
    { 'A', 'S', 'e', 'c', 'u', 'r', 'e', 'S', 'e', 'c', 'r', 'e', 't', 'K', 'e', 'y' };
    // Performs Encryption
    public static String encrypt99(String plainText) throws Exception
    {
        Key key = generateKey();
        Cipher chipper = Cipher.getInstance(algorithm);
        chipper.init(Cipher.ENCRYPT_MODE, key);
        byte[] encVal = chipper.doFinal(plainText.getBytes());
        String encryptedValue = new BASE64Encoder().encode(encVal);
    }
}

```



```

        return encryptedValue;
    }

    // Performs decryption
    public static String decrypt(String encryptedText) throws Exception
    {
        // generate key
        Key key = generateKey();
        Cipher cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.DECRYPT_MODE, key);
        byte[] decodedValue = new
BASE64Decoder().decodeBuffer(encryptedText);
        byte[] decValue = cipher.doFinal(decodedValue);
        String decryptedValue = new String(decValue);
        return decryptedValue;
    }

    //generateKey() is used to generate a secret key for AES algorithm
    private static Key generateKey() throws Exception
    {
        Key key = new SecretKeySpec(keyValue, algorithm);
        return key;
    }
}

// DATA ENCRYPTION CODE

public class Encryptdata {
    public static String encrypt(String Data) throws Exception
    {
        System.out.println("Encrypted in coming");
        Key key = PublicKey.generateKey();
        Cipher c = Cipher.getInstance("AES");
        c.init(Cipher.ENCRYPT_MODE, key);
        byte[] encVal = c.doFinal(Data.getBytes());
        String encryptedValue = new BASE64Encoder().encode(encVal);
        return encryptedValue;
    }
}

```



```

        public static String decrypt(String encryptedData) throws Exception
        {
            Key key = PublicKey.generateKey();
            Cipher c = Cipher.getInstance("AES");
            c.init(Cipher.DECRYPT_MODE, key);
            byte[] decodedValue = new
BASE64Decoder().decodeBuffer(encryptedData);
            byte[] decValue = c.doFinal(decodedValue);
            String decryptedValue = new String(decValue);
            //System.out.println("key is :::"+ABEKey.generateKey());
            return decryptedValue;
        }
    }

```

// DATA TRANSFER CODE

```
package Imple;
```

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import Bean.managebean;
import Bean.sregbean;
import Bean.taskbean;
import Bean.tlfilebean;
import Bean.tlregbean;
import Dbconn.dbconn;
import Inter.Inter;

```

```
public class Imple implements Inter {
```

```

    Connection con;
    @Override
    public int tlreg(tlregbean tb) {
        // TODO Auto-generated method stub
        int reg=0;
        con=dbconn.create();
        try {

```



```

        PreparedStatement ps=con.prepareStatement("INSERT
        INTO bc04.tlregister VALUES(id,?,?,?,?,?,?,?)");
        ps.setString(1, tb.getName());
        ps.setString(2, tb.getEmail());
        ps.setString(3, tb.getMobile());
        ps.setString(4, tb.getPassword());
        ps.setString(5, tb.getCpassword());
        ps.setString(6, tb.getImage());
        ps.setString(7, tb.getTeam());
        reg=ps.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return reg;
}

@Override
public boolean tlog(tlregbean tb) {
    // TODO Auto-generated method stub
    boolean log=false;
    con=dbconn.create();
    try {
        PreparedStatement ps=con.prepareStatement("SELECT *
        FROM `bc04`.`tlregister` where team=? and email=? and password=?");
        ps.setString(1, tb.getTeam());
        ps.setString(2, tb.getEmail());
        ps.setString(3, tb.getPassword());
        ResultSet rs=ps.executeQuery();
        log=rs.next();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```



```

        return log;
    }
    @Override
    public int tlf(tlfilebean tf) {
        // TODO Auto-generated method stub

        int dd=0;
        try
        {
            con=dbconn.create();
            PreparedStatement ps=con.prepareStatement("INSERT INTO
bc04.tlfileadd VALUES(id,?,?,?,?,?,?,?,?,?,?)");
            ps.setString(1, tf.getTitle());
            ps.setString(2, tf.getDescription());
            ps.setString(3, tf.getFilename());
            ps.setString(4, tf.getFiletype());
            ps.setString(5, tf.getFilesize());
            ps.setString(6, tf.getFilecontent());
            ps.setString(7, tf.getEncrypt());
            ps.setString(8, tf.getDecrypt());
            ps.setString(9, tf.getTlmail());
            ps.setString(10, tf.getTeam());
            dd=ps.executeUpdate();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        return dd;
    }
    @Override
    public int sreg(sregbean sb) {
int reg=0;
        con=dbconn.create();
        try {

```



```

        PreparedStatement ps=con.prepareStatement("INSERT
        INTO bc04.staffreg VALUES(id,?,?,?,?,?,?,?)");
        ps.setString(1, sb.getName());
        ps.setString(2, sb.getEmail());
        ps.setString(3, sb.getMobile());
        ps.setString(4, sb.getPassword());
        ps.setString(5, sb.getCpassword());
        ps.setString(6, sb.getImage());
        ps.setString(7, sb.getTeam());
        reg=ps.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return reg;
}

@Override
public boolean slog(sregbean sl) {
    boolean log=false;
    con=dbconn.create();
    try {
        PreparedStatement ps=con.prepareStatement("SELECT *
        FROM `bc04`.`staffreg` where team=? and email=? and password=?");
        ps.setString(1, sl.getTeam());
        ps.setString(2, sl.getEmail());
        ps.setString(3, sl.getPassword());
        ResultSet rs=ps.executeQuery();
        log=rs.next();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return log;
}
}

```



```

@Override
public int task(taskbean t) {
    int dd=0;
    try
    {
        con=dbconn.create();
        PreparedStatement ps=con.prepareStatement("INSERT INTO
bc04.task VALUES(idtask,?,?,?,?,?,?,?,?)");
        ps.setString(1, t.getTask());
        ps.setString(2, t.getDes());
        ps.setString(3, t.getFilename());
        ps.setString(4, t.getTlemail());
        ps.setString(5, t.getTeam());
        ps.setString(6, t.getTlkey());
        ps.setString(7, "Request");
        ps.setString(8, t.getStaff());
        dd=ps.executeUpdate();
    }

    catch(Exception e)
        e.printStackTrace();
    return dd;
}

@Override
public int upk(taskbean u) {
    int update=0;
    try
    {
        con=dbconn.create();
        PreparedStatement ps = con.prepareStatement("UPDATE
bc04.task SET status='"+u.getStatus()+"' where idtask='"+u.getId()+"");
        update=ps.executeUpdate();
    }
    catch(Exception e)
    {

```



```

        e.printStackTrace();
    }
    return update;
}
@Override
public int mkey(managebean mb) {
    int dd=0;
    try
    {
        con=dbconn.create();
        PreparedStatement ps=con.prepareStatement("INSERT INTO
bc04.manageract VALUES(id,?,?,?,?,?,?,?,?,?,?)");
        ps.setString(1, mb.getTask());
        ps.setString(2, mb.getFilename());
        ps.setString(3, mb.getTlemail());
        ps.setString(4, mb.getTeam());
        ps.setString(5, mb.getTkey());
        ps.setString(6, mb.getSemail());
        ps.setString(7, mb.getMkey());
        ps.setString(8, mb.getPhash());
        ps.setString(9, mb.getAhash());
        ps.setString(10, mb.getMail());

        ps.setString(11, "sended");
        dd=ps.executeUpdate();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    return dd;
}
}

```

// USER INTERFACE CODE


```

<!DOCTYPE html>
<html lang="en">

<!-- dustech/about.html 13 Nov 2019 12:55:21 GMT -->
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="author" content="irstheme">

  <title> Dustech - Industry & Factory HTML Template </title>

  <link href="assets/css/themify-icons.css" rel="stylesheet">
  <link href="assets/css/flaticon.css" rel="stylesheet">
  <link href="assets/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/css/animate.css" rel="stylesheet">
  <link href="assets/css/owl.carousel.css" rel="stylesheet">
  <link href="assets/css/owl.theme.css" rel="stylesheet">
  <link href="assets/css/slick.css" rel="stylesheet">
  <link href="assets/css/slick-theme.css" rel="stylesheet">
  <link href="assets/css/swiper.min.css" rel="stylesheet">
  <link href="assets/css/odometer-theme-default.css" rel="stylesheet">
  <link href="assets/css/owl.transitions.css" rel="stylesheet">
  <link href="assets/css/jquery.fancybox.css" rel="stylesheet">
  <link href="assets/css/style.css" rel="stylesheet">

  <a class="navbar-brand" href="index-2.html"></a>
</div>
<div id="navbar" class="navbar-collapse collapse navigation-holder">
  <button class="close-navbar"><i class="ti-close"></i></button>
  <ul class="nav navbar-nav">
    <li class="menu-item-has-children">
      <a href="#">Home</a>
      <ul class="sub-menu">
        <li><a href="index-2.html">Home Default</a></li>
        <li><a href="index-3.html">Home style 2</a></li>
        <li><a href="index-4.html">Home style 3</a></li>
      </ul>
    </li>
    <li><a href="about.html">About</a></li>
    <li class="menu-item-has-children">
      <a href="#">Pages</a>
      <ul class="sub-menu">
        <li><a href="about.html">About</a></li>
        <li><a href="contact.html">Contact</a></li>
        <li><a href="testimonials.html">Testimonials</a></li>
        <li><a href="testimonials-s2.html">Testimonials style 2</a></li>
        <li><a href="team.html">Team</a></li>
        <li><a href="faq.html">FAQ</a></li>
        <li><a href="404.html">404</a></li>
        <li><a href="shop.html">Shop</a></li>
      </ul>
    </li>
  </ul>
</div>

```



```

        <li><a href="shop-single.html">Shop single</a></li>
    </ul>
</li>
<li class="menu-item-has-children">
    <a href="#">Services</a>
    <ul class="sub-menu">
        <li><a href="services.html">Services</a></li>
        <li><a href="services-s2.html">Services style 2</a></li>
        <li><a href="service-single.html">Mechanical
Engineering</a></li>
        <li><a href="oil-gas.html">Oil And Gas</a></li>
        <li><a href="power-energy.html">Power And Energy</a></li>
        <li><a href="sanitary-plumbing.html">Sanitary &
Plumbing</a></li>
        <li><a href="electrical-power.html">Electrical Power</a></li>
    </ul>
</li>
<li class="menu-item-has-children">
    <a href="#">Projects</a>
    <ul class="sub-menu">
        <li><a href="portfolio.html">Projects</a></li>
        <li><a href="portfolio-s2.html">Projects style 2</a></li>
        <li><a href="project-single.html">Project
    <p>Whole of her lower arm towards the viewer. Gregor then
turned to look out the window</p>
    </div>
    <div class="grid">
        <i class="fi flaticon-circular-label-with-certified-stamp"></i>

<!-- start site-footer -->
<footer class="site-footer">
    <div class="upper-footer">
        <div class="container">
            <div class="row">
                <div class="col col-lg-3 col-md-3 col-sm-6">
                    <div class="widget about-widget">
                        <div class="logo widget-title">
                            
                        </div>
                        <p>Mikago arm towards the viewer gregor then turned to look out
the window at the dull weather</p>
                    <div class="social-icons">
                        <ul>
                            <li><a href="#"><i class="ti-facebook"></i></a></li>
                            <li><a href="#"><i class="ti-twitter-alt"></i></a></li>
                            <li><a href="#"><i class="ti-linkedin"></i></a></li>
                            <li><a href="#"><i class="ti-pinterest"></i></a></li>
                            <li><a href="#"><i class="ti-vimeo-alt"></i></a></li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```



```

<div class="col col-lg-3 col-md-3 col-sm-6">
  <div class="widget link-widget">
    <div class="widget-title">
      <h3>Usefull Links</h3>
    </div>
    <ul>
      <li><a href="#">About us</a></li>
      <li><a href="#">Our services</a></li>
      <li><a href="#">Contact us</a></li>
      <li><a href="#">Meet team</a></li>
    </ul>
    <ul>
      <li><a href="#">Provacu Policy</a></li>
      <li><a href="#">Testimonials</a></li>
      <li><a href="#">News</a></li>
      <li><a href="#">FAQ</a></li>
    </ul>
  </div>
</div>
<div class="col col-lg-3 col-md-3 col-sm-6">
  <div class="widget contact-widget service-link-widget">
    <div class="widget-title">
      <h3>Our Address</h3>
    </div>
    <ul>
      <li>25/19 Duel aveniew, new Booston town, Ghana</li>
      <li><span>Phone:</span> 84667441</li>
      <li><span>Email:</span> demo@example.com</li>
      <li><span>Office Time:</span> 10AM- 5PM</li>
    </ul>
  </div>
</div>
<div class="col col-lg-3 col-md-3 col-sm-6">
  <div class="widget newsletter-widget">
    <div class="widget-title">
      <h3>Newsletter</h3>
    </div>
    <p>You will be notified when somthing new will be appear.</p>
    <form>
      <div class="input-1">
        <input type="email" class="form-control" placeholder="Email
Address *" required>
      </div>
      <div class="submit clearfix">
        <button type="submit"><i class="ti-email"></i></button>
      </div>
    </form>
  </div>
</div>
</div> <!-- end container -->
</div>

```



```
<div class="lower-footer">  
    <div class="container">  
        <div class="row">  
            <div class="separator"></div>  
            <div class="col col-xs-12">  
                <p class="copyright"><a href="templateshub.net">Templates  
Hub</a></p>  
            <div class="extra-link">  
                <ul>  
                    <li><a href="#">Privace & Policy</a></li>  
                    <li><a href="#">Terms</a></li>  
                    <li><a href="#">>About us</a></li>  
                    <li><a href="#">FAQ</a></li>  
                </ul>  
            </div>  
        </div>  
    </div>  
</div>  
<!-- end site-footer -->  
</div>  
<!-- end of page-wrapper -->
```


C. RESEARCH PAPER

SECURED FILE SHARING SYSTEM USING CRYPTOGRAPHY AND BLOCKCHAIN

Diganta Panda

Department of IT, Sathyabama Institute of
Science and Technology, Chennai, India
digantapandadp@gmail.com

Raushan Kumar

Department of IT, Sathyabama Institute of
Science and Technology, Chennai, India
raushanlive2000@gmail.com

Dr. G Mathivanan, M.E., Ph.D

Associate Professor

Department of IT, Sathyabama Institute of
Science and Technology, Chennai, India

Abstract

In today's world data is the most important thing. From small enterprises to trillion-dollar multinational companies, data is the most important resource and is worth billions of dollars. So its security and privacy becomes essential and uncompromisable. But nowadays hackers and spammers find vulnerabilities and steal the private data and misuse it by selling them in dark web etc. Some employees also share valuable data through WhatsApp which is very risky. To deal this, our paper proposes a system which uses combination of blockchain and cryptography to make those highly safe and secured and strictly prohibits any third or unwanted party from accessing the data. It uses SHA 256 algorithm for cryptographic security. It uses AES algorithm for encryption-decryption. It uses symmetric key for privacy. We have added another layer of security by implementing QR code scanning. Ultimately, staff members, team leaders and higher officials like managers or senior executives can share data securely. No hacker can breach the system. Our work can be implemented in other domains as well like government organizations, chatting, business deals, financial transactions, online receipt, interpersonal lives of ordinary people thus making it realistic and very practical.

Keywords—Encryption, Decryption, Cryptography, Blockchain, Hashing, AES algorithm, SHA 256 algorithm

INTRODUCTION

Private and sensitive information must be kept secret and secure from any outsider who wants to access it. They must have the authentication to do so. Data security is very important here. Appropriate measures and security features have to be there to prevent from any hacking or database breaching etc. Private data or sensitive information is the important data that needs to be well guarded from third party persons or outsiders.

Data can be protected and secured when data encryption benefits a business. A technology called encryption uses complicated algorithms to hide data. A cypher key will be provided to users who are encrypting their data. This is a unique key that unlocks the data, making the content readable once more. Naturally, this key should only be accessible to you and other people who have been given permission to do so; otherwise, encryption would be useless.

Your concern for privacy will decrease if you encrypt your data. Your sensitive information will remain private thanks to encryption.

There are several importance of data encryption. Every business needs encryption today because it gives them the ability to safeguard sensitive information by converting it to ciphertext, which cannot be deciphered without the use of an encryption key. It is known as "encoding" to do this. Due to the fact that only those in possession of an encryption key can decipher the data and reveal the true information, encryption makes it nearly impossible for cybercriminals or other unauthorized parties to steal and misuse the data.

Because data breaches happen to even the largest corporations with the highest cybersecurity budgets, the significance of encryption cannot be overstated. Having said that, there is a chance that your data could be compromised even if it is in a secure infrastructure. But even if they were stolen, your files could be that much more impenetrable with data encryption.

So, how would this operate in practice? Imagine that a user of a company's data sent some private documents via email, which is typically not encrypted in any way. Because hackers may intercept these emails and obtain access to private information, this information is put at risk.

However, even if the data were intercepted and stolen, the cyber attacker would not be able to access the data because it would be in an unreadable format if the user used an encrypted file sharing solution like Titan File instead. The information would need to be successfully decrypted in order for anyone without the encryption key to access it.

1.1. Advanced Encryption Standard (AES)

AES stands for Advanced Encryption Standard and is a majorly used symmetric encryption algorithm. It is mostly utilized to encrypt and safeguard electronic data. AES is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. Because it is faster and more effective than DES (Data encryption standard), it was used to replace DES. Three block cyphers make up the AES encryption system, which is used to encrypt data.

It generates 128 bits of encrypted cypher text as output from 128 bits of input. AES operates utilising a chain of linked operations that replace and shuffle the input data, which is known as the substitution-permutation network principle. Two Belgian cryptographers, Joan Daemen and Vincent Rijmen, created the AES variation of the Rijndael block cipher and presented a proposal to NIST as part of the AES

selection process. A family of cyphers called Rijndael has various key and block sizes. NIST chose three Rijndael family members for AES, each having a 128-bit block size but three distinct key lengths: 128, 192, and 256 bits. It is adaptable and has software and hardware implementations. It offers excellent security and can stop many attacks. Since it is copyright-free, anyone can use it anywhere in the world. For 128-bit keys, there are 10 rounds of processing.

There are several **advantages** of using AES algorithm. It can be used with both software and hardware. It offers users a high level of security. It offers one of the top encryptions open-source solutions. This algorithm is very reliable and robust.

Few **disadvantages** are there as well. It requires many rounds for encryption. It is hard to implement on software. It needs much processing at different stages. It is difficult to implement when performance has to be considered.

The AES algorithm's key lengths—128, 192, and 256—have been designed and tested to be strong enough to protect sensitive data higher than or equal to the SECRET level. It will be necessary to use either the 192 or 256 key lengths for TOP SECRET data. Prior to being purchased and used, AES implementation in products meant to safeguard national security systems and/or information must be examined and certified by the NSA.

1.2 Hashing

Hashing is the process of scrubbing raw data until it can no longer be restored to its original form. It takes certain part of information and runs it through a function that performs mathematical operations on the plaintext. This function is known as the hash function, and the output is termed as the hash value.

The hash function converts the plaintext to its respective hash value. It is non-reversible. It means hash value can't be converted back to its original plaintext. No matter how many times they are used, hash functions always produce the same result when the input is left untouched.

1.3 SHA 256 (Secure Hash Algorithm)

SHA, which stands for Secure Hash Algorithm, is a family of algorithms that includes SHA 256. In 2001, the NSA and NIST collaborated and introduced it as an advancement for the SHA 1 family, which was gradually becoming less resistant to brute force attacks. The 256 in the name refers to the final hash value, meaning that regardless of the amount of plaintext present, the hash value will always be 256 bits. The other SHA family algorithms are somewhat comparable to SHA 256.

There are certain characteristics of SHA 256 algorithm:

- **Length of the Message:** The cleartext should be no longer than 264 bits. To keep the digest as random as possible, the size must be in the comparison area.
- **Digest Length:** The length of the hash digest for the SHA-256 algorithm should be 256 bits, for the SHA-512 algorithm 512 bits, and so forth. Larger digests typically imply a lot more calculations at the expense of speed and storage.
- **Irreversible:** All hash functions, including the SHA 256, are intended to be irreversible. If you already have the digest, you shouldn't receive the plaintext, and if you run the digest through the hash function once more, it shouldn't return the original value.

Many applications use the SHA algorithm, some of which include the following,

- **Digital signature verification:** To confirm the legitimacy of a document or file, digital signatures use asymmetric encryption methodology. To a large extent, hash algorithms like SHA 256 ensure that the signature is verified.
- **Password Hashing:** Websites store user passwords in a hashed format, for two reasons. Due to the similar size of all the digests, it promotes privacy and lightens the burden on the main database.
- **SSL Handshake:** Using SHA functions, the SSL handshake is a crucial part of web browsing sessions. To set up a secure connection, your web browsers and the web servers must agree on encryption keys and hash authentication.
- **Integrity checks:** As was mentioned above, the SHA 256 algorithm and the MD5 algorithm have been used to verify file integrity. It ensures that files were not altered in transit and helps maintain the full functionality of files.

LITERATURE SURVEY

1) "Secure Computation by Secret Sharing using Input Encrypted with Random Number", Kenichi Nakamura and Sayed Mohammad Kashif and Mohammad Tabrez aalam, 2019

When $n \geq 2k + 1$, it is typically thought that uncompromisable safe and secured processing using n (a, k) limit secret sharing is not possible. In order to achieve secure programming using n (a, k) limit privacy sharing, we started off with the technique of identifying the conditions necessary for safe programming under the rule of a $2k + 1$. We then

demonstrated this under the following three preconditions. Additionally, we describe the benefit of providing secure computations with secret data that has been encrypted using a random number. The speeding up of computation time is one of the benefits. As a result, in comparison to previous techniques, we achieve a quicker online phase for computations like inner product operations.

2) "CNN Based Malicious Website Detection by Invalidating Multiple Web Spams", Gonzhie Liuzhen And Jongich-Hier Leek, 2020

Although many methods for identifying malicious websites have been put forth, it is getting harder and harder for those techniques to do so in the modern era. With a variety of web fraud methods, many malicious websites can continue to go undetected. We've outlined three categories of Web spam tactics used by malicious websites in this paper, including redirection spam, hidden IFrame spam, and content hiding spam. Then, in order to disprove Web spam, we introduce a brand-new detection technique that adopts the viewpoint of users and screenshots malicious websites. Convolutional neural networks, a subset of deep neural networks, are used as the classification algorithm in the suggested detection technique. Two different experiments have been carried out to confirm the method's efficacy. First, a constructed complex dataset was used to test the proposed method. Results of comparisons between the suggested method and typical machine learning-based detection algorithms are presented. Second, for three months, the suggested technique was tested in a real-world Web environment to identify malicious websites. These test results show that the suggested approach performs better and can be used in a real-world Web environment.

3) "Secure multi-party computation in differential private data with Data Integrity Protection", Sundari S, Ananthi M, 2015

Nowadays, when data are shared among various parties, secure multiparty computation (SMC) is required. Additionally, organisations are encouraged to work together with other parties who run the same business for the benefit of both parties. Users can obtain a lot of information from larger data sets using SMC without having to reveal the data themselves. In this project, the methods of differential privacy for vertically partitioned data between parties and secure multi-party computation are combined. A multi-party protocol for the exponential mechanism has been proposed to accomplish this. For the majority of computer applications and data servers, dependable access to data is essential. Unauthorized access to

stored data is caused by a few factors. While integrating the data in multiparty computation, Two Phase Validation (2PV) offers user authentication. Some errors may cause data to become corrupt. Disk errors are widespread today, but storage technologies are not built to handle them. A straightforward integrity violation is discovered by higher level software, which results in additional data loss. The proposed system uses Distributed Data Integrity (DDI) Protection to check the integrity of random subsets of data against accidental or intentional corruptions.

4) **"White-Box Traceable Ciphertext-Policy Attribute-Based Encryption in Multi-Domain Environment"**, Yongle Tang, Xu He, Yi Chang Shek, 2020

The decryption keys in a ciphertext based policy attribute system encryption (CP-ABE) system only relate to shared data from multiple members and does not include any personal data about their original owners. Therefore, if a decryption key is exposed, there is no practical way to identify the suspect user. The user tracing issue is now preventing the practical application of CP-ABE. Some traceable/accountable CP-ABE schemes have been established in order to address it. However, there will be new difficulties when trying to solve the user tracing issue in a multi-domain environment. Domains and intradomain users make up the two layers of a multi-domain environment. It is best practise to trace a user's location in a domain before tracing the user within the domain. This proposed scheme has relatively low asymptotic communication and computation costs when compared to similar schemes. These benefits make the suggested solution more workable for resolving the user tracing issue in a multi-domain setting.

5) **"A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing"**, Guefeng Lin And Himanshu Gulia, 2017

CP-ABE, or ciphertext based policy attribute system encryption, is a great cryptographic method for specifically controlling the accessibility to cloud-hosted data that has been outsourced. Key management has some disadvantages, though, which limit its use. We show that clients run the risk of key exposure because front-end devices, such as smart phones, typically offer only rudimentary privacy protection, which makes it difficult to detect but always present in earlier research. The practical application of ABE is further restricted by the significant client decryption overhead. Supportive

key management protocol in CP-ABE in this paper has been presented. Our design achieves distributed private key generation, issuance, and storage without the need for additional infrastructure. For key updation, instant attribute cancellation is offered. The suggested technique successfully resolves both important exposure and important escrow issues. In the meantime, it significantly lowers client decryption overhead. Our scheme performs a little bit better than other representative CP-ABE schemes. Finally, we demonstrate the proposed protocol's security.

6) **"Efficient Compressed Ciphertext Length Scheme Using Multi-Authority CP-ABE for Hierarchical Attributes"**, Liyong Chang And Chang Ling, 2018

While encrypting and decrypting, an encryption based on it allows the people using it to get known. The current attribute-based encryption techniques have discovered that the encryption joining technique does not have high calculation productivity and ciphertext data repetition if the user's downloading method contains significant amount of valuable data labelled. So, to fix these issues, we have suggested in this paper a tree like multi-authority attribute-based encryption on prime order groups. Our encryption method uses a polycentric attribute authorization system that is based on an AND gate access structure. Each attribute power starting from the project makes a unified attribute index like structure.

7) **"A systematic literature review of blockchain cyber security"**, Paul J. Taylor, Tooska Dargahi, Ali Dehghantanha, Reza M. Parizi, Kim-Kwang Raymond Choo., 2020

Since the publication of Satoshi Nakamoto's white paper on Bitcoin in 2008, blockchain has (slowly) become one of the most frequently discussed methods for securing data storage and transfer through decentralized, trustless, peer-to-peer systems. Our findings show that the Internet of Things (IoT) lends itself well to novel blockchain applications, as do networks and machine visualization, public key cryptography, web applications, certification schemes and the secure storage of Personally Identifiable Information (PII). To achieve this, a multi-party protocol has been proposed for the exponential mechanism. Reliable access to data is must for most computer applications and data servers. This timely systematic review also sheds light on future directions of research, education and practices in the blockchain and cyber security space, such as security of blockchain in IoT, security of blockchain for AI data, and sidechain security etc.

INFERENCES FROM LITREATURE SURVEY

From the above-mentioned literature works, it is evident that there has been surplus research on various. It is clear that all of them have their specific pros and cons.

Few of the researches have used RSA or DES algorithm for encryption which is lesser effective. And some of them have used SHA1 tool whose updated version has arrived.

Few publications are very secure but few have some drawbacks too by which hackers can break the integrity of the system and can access the secret data.

EXISTING SYSTEM

In previous projects related to file sharing, various techniques and algorithms have been used. Ciphertext-policy attributebased encryption (CP-ABE) technology, based on the linear secret sharing scheme (LSSS), an efficient scheme is proposed to realize a collaborative decryption function. For any user group, when the user's attribute set cannot access the ciphertext alone, the private key of other users in the same group can be used for collaborative decryption with the permission of the data owner. Other projects have used RSA, DES algorithm for encryption.

PROPOSED SYSTEM AND ARCHITECTURE

In our project, we have used java as our basic programming language as it is highly reliable and efficient compared to its counterparts. Cryptography usage is there like AES algorithm for encryption and decryption. For front-end to make our user interface we have used HTML, CSS, JavaScript, bootstrap along with jsp. SHA 256 algorithm which is a blockchain tool is used for the additional security from hackers.

There are several modules which explains how our system works step by step.

1. Staff Register:

Here the staff registers himself when he is not known to the database. The staff is asked to write his name, email id, phone no, the team which he belongs to, and lastly he is told to create the password for logging in in future.

2. Staff Login:

After the staff registers or any existing staff is there, they need to login by filling their credentials.

3. Staff File View:

In this part the staff member will be able to view the file shared by the team leader.

4. Staff File Request:

In this part, the staff member will request the team leader to allow him to view and download the file and a request will go to team leader's organization portal.

5. Staff File Download:

In this module the staff download the file after management and team leader accepts the request. It will be stored on local storage.

6. Team Leader Login:

In this part, the team leader will login into the system and the database. Team leader has all the important organization sensitive documents.

7. Team Leader File Upload:

The team leader will upload the file which will be shared to staff member. It is the sensitive information.

8. Team Leader File View:

In this module the shared file goes to the database and gets encrypted. It is all done using asymmetric encryption of AES algorithm. Team leader can clearly view the file he has shared for the staff member to access.

9. Management Login:

Management are the executive level personals in the organization e.g., Board of director, managing director, manager, CEO etc. In this part, the management needs to login by filling their credentials by proving their authenticity.

10. Management Team Leader Registration:

In this part, the management and team leader registers themselves. They prove their identity by putting their credentials and thus can access the database of the company and can share files and sensitive data and information.

11. Management Generate Key:

In this part the management team generates the secret key for the request made by staff for accessing the file. Key is for security purpose. After the staff receives the key, finally the staff can download the file.

12. Management Response:

In this part the management's response is recorded. In the ledger every sensitive transaction is recorded. It

uses the concepts of blockchain like previous and current hash along the shared data file.



Fig: System Architecture Diagram

Any member of the company or the organisation can login. Team leader can share the file or the sensitive information to the staff along with the permission of the managing authority in the company. Upon approval from team leader and the management, staff will finally be able to access the database and download that particular file or piece of information. For additional security no same file could be shared again. For another layer of security QR code will get generated which the staff member will scan and get the code and can finally download.

COMPARISON OF EXISTING AND PROPOSED

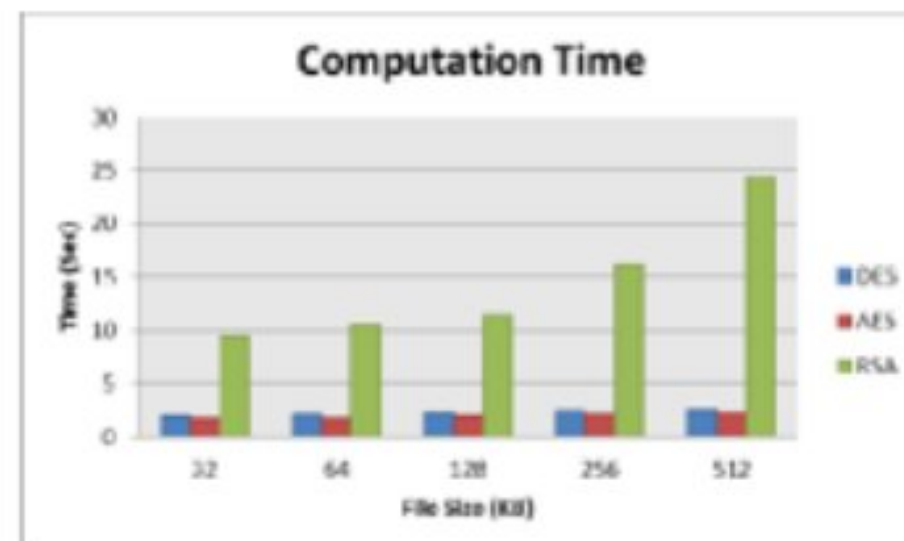


Fig: Comparison of AES with other algorithms

AES is far safer and more secured than DES (Data Encryption Standard). DES can be attacked by hackers easily and is vulnerable to brute force attacks.

128 bits of plaintext can be encrypted by AES but in case of DES its only 64 bits. Unlike DES, AES is compatible with hardware as well as software but in DES it restricts to only hardware.

AES is more advanced than RSA and is technically extensive than it. It can encrypt high amounts of data. Based upon the length of key which encrypts decrypts the data, SHA 1 and SHA 2 is differentiated. SHA2 is much more advanced and secured than SHA1.

CONCLUSION

Information responsiveness concerns data that ought to be shielded from unapproved access or revelation because of its delicate nature. For some's purposes, that may be Group pioneer, Staff subtleties records. Delicate information is private data that should be remained careful and far away from all outcasts except if they have consent to get to it. Admittance to delicate information ought to be restricted through adequate information security and data security rehearses intended to forestall.

FUTURE ENHANCEMENTS:

1. To maximize the amount of data size and operation speed of our project.
2. Implement it in real world by putting it in web servers like AWS and selling it to clients.

REFERENCES

- [1] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147-156, 2020.
- [2] D. Liu and J. Lee, "CNN based Malicious Website Detection by Invalidating Multiple Web Spams," *IEEE Access*, vol. 8, no. 1, pp. 97258-97266, 2020.
- [3] W. Martin, V. Friedhelm, and K. Axel, "Tracing manufacturing processes using blockchain-based token compositions," *Digital Communications and Networks*, vol. 6, no. 2, pp. 167-176, 2019.
- [4] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6-14, 2018.
- [5] L. Peng, W. Feng, and Z. Yan. (2020). Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks*. [Online]. Available: <https://doi.org/10.1016/j.dcan.2020.05.008>.

[6] N. Kakade and U. Patel, "Secure Secret Sharing Using Homomorphic Encryption," in *Proc. 2020 11th International Conference on Computing, Communication and Networking Technologies*, 2020, pp. 1-7.

[7] S. Sundari and M. Ananthi, "Secure multi-party computation in differential private data with Data Integrity Protection," in *Proc. 2015 International Conference on Computing and Communications Technologies*, 2015, pp. 180-184.

[8] S. Jiao, T. Lei, Y. Gao, Z. Xie and X. Yuan, "Known-Plaintext Attack and Ciphertext-Only Attack for Encrypted Single-Pixel Imaging," *IEEE Access*, vol. 7, no.2, pp. 119557-119565, 2019.

[9] S. Kaushik, and S. Puri, "Online transaction processing using enhanced sensitive data transfer security model," in *Proc. 2012 Students Conference on Engineering and Systems*, 2012, pp. 1-4.

[10] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li and R. Chen, "NutBaaS: A Blockchain-as-a-Service Platform," *IEEE Access*, vol. 7, pp. 134422-134433, 2019.

[11] F. Casino and C. Patsakis, "An Efficient Blockchain-Based Privacy-Preserving Collaborative Filtering Architecture," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1501-1513, Nov. 2020.

[12] D. Chkhaev, J. Hooman and P. van der Stok, "Mechanical verification of transaction processing systems," in *Proc. ICFEM 2000. Third IEEE International Conference on Formal Engineering Methods*, 2000, pp. 89-97.

[13] S. Zhang, and J. H. Lee, "Mitigations on Sybil-based Double-spend Attacks in Bitcoin," *IEEE Consumer Electronics Magazine*, vol.7, no. 2, pp. 1-1, 2020.

[14] X. Wang, Q. Feng and J. Chai, "The Research of Consortium Block chain Dynamic Consensus Based on Data Transaction Evaluation," in *Proc. 2018 11th International Symposium on Computational Intelligence and Design*, 2018, pp. 214-217.

[15] S. Zhang, and J. H. Lee, "A group signature and authentication scheme for blockchain-based mobile-edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, 4557-4565, 2019.