# A

**MAJOR PROJECT ON**

# BLOCK-CHAIN BASED DATA STORAGE WITH PRIVACY AND AUTHENTICATION IN INTERNET OF THINGS

Submitted in partial fulfilment of the degree of the Bachelor of Technology in

# COMPUTER SCIENCE AND ENGINEERING

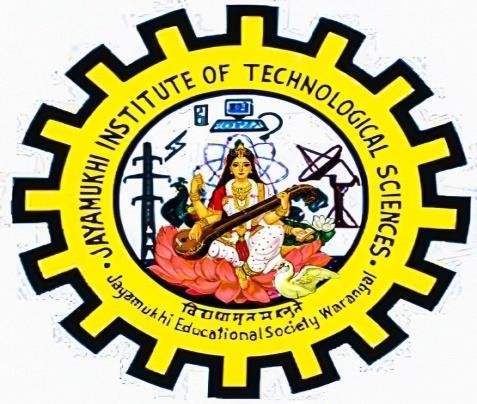
## Mohammad abdul shoaib 19C41A05C9

N B Shashidhar 19C41A05C8

## Naseerhuddhin 19C41A05D0

Mohammad salman areeb 19C41A05C4 Under the guidance of

**DR.P.SRINIVAS RAO**



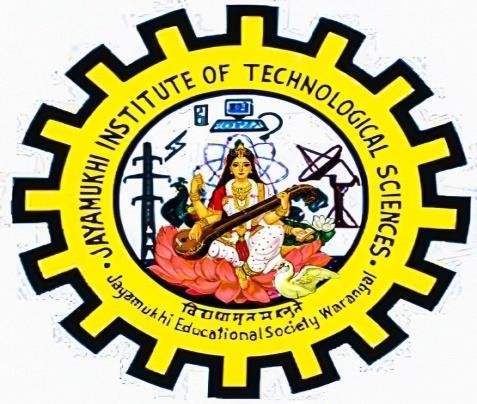
### Department of Computer Science and Engineering

**JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES NARSAMPET, WARANGAL-506 332**

**(Autonomous, Affiliated to JNTUH, Accredited by NBA, NAAC-A) January 2023**

**JAYAMUKHI INSTITUTE OF TECHNOLOGICAL SCIENCES NARSAMPET, WARANGAL – 506 332**

**(Autonomous, Affiliated to JNTUH, Accredited by NBA, NAAC-A) CERTIFICATE**



This is to certify that Major project Report entitled **“BLOCKCHAIN BASED DATA STORAGE WITH PRIVACY AND AUTHENTICATION IN INTERNET OF THINGS**” is a bonafide work of the student Mohammad abdul shoaib (19C41A05C9) , N B Shashidhar (19C41A05C8), Naseerhuddhin (19C41A05D0), Mohammad salman Areeb (19C41A05C4) submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING during the academic year 2022-2023

### Guide Head of the department

**principal**

**ACKNOWLEDGMENT**

#### We express our immense gratitude and sincere thanks to **Dr. V. JANAKI**, principal, Jayamukhi Institute of Technological Sciences for her constant motivation and support.

We express our gratitude and sincere thanks to **DR. P. SRINIVAS RAO**, Head of the Department Of CSE, Jayamukhi Institute of Technological Sciences for his constant Motivation and support.

#### We are grateful to our Project Guide **DR.P.SRINIVAS RAO** , Assistant Professor for his Valuable guidance and giving us the opportunity to do our work in a better manner.

We are also thankful to our Management for providing all the facilities in completing the project.

# ABSTRACT

Internet of Things (IoT) composed of large number of sensing devices with a variety of features applicable for various applications. In such scenarios, due to low data handling capabilities, limited storage, and security aspects, it is quite challenging to protect networks against illegal information access and utilizes storage efficiently. Though researchers provide various solutions for security and data storage, but a few solutions are appropriate for wireless sensor networks (WSNs)-enabled IoTs. Therefore, a blockchain-based decentralized framework integrated with authentication and privacy- preserving schemes is developed for the secure communication in WSNs-enabled IoTs. Registration, certification, and revocation process are employed for the communication with sensor nodes and base station (BS) in a cloud computing environment. In this scheme cluster heads forward the collected information to the BS. Consequently, BS records all the key parameters on the distributed blockchain and large data is forwarded to clouds for the storage. The revoked certificates of all malicious nodes are eliminated from blockchain by BS. The performance of the proposed scheme is scrutinized in terms of detection accuracy, certification delay, computational, and communicational overheads. The simulated results, comparative analysis, and security validation support the superiority of the proposed solution overthe existing approaches.

**Index Terms**—Blockchain, distributed, Internet of Things (IoT), privacy and authentication, security, wireless sensor network (WSN).

TABLE OF CONTENTS

|  |  |
| --- | --- |
| ABSTRACT | PageNo.  i |
| TABLE OF CONTENTS | ii |
| 1.INTRODUCTION | 1 |
| 1.1 Literature Survey | 2 |
| 1.2System Analysis | 5 |
| 1.3 System Requirements | 7 |
| 1.4 System Study | 8 |
| 2. SYSTEM DESIGN | 11 |
| 2.1 System Architecture | 12 |
| 2.2 UML Diagrams | 13 |
| 3. MODULE DIVISION | 19 |
| 3.1 Input/Output Design | 20 |
| 3.2 Software Development Tools | 21 |
| 4. SOFTWARE TESTING | 43 |
| 4.1 Unit Testing | 43 |
| 4.2 Performance Testing | 43 |
| 4.3 Integration Testing | 44 |
| 4.4 Test Cases | 44 |
| 5. EXPERIMENTAL RESULTS | 49 |
| 6. CONCLUSION AND FUTURE SCOPE | 52 |
| 7. BIBLIOGRAPHY | 54 |

# INTRODUCTION

In current era, Internet of Things (IOT) is one of the most popular, useful, and dominant technologies in wireless communication and information processing. IoT is the formulation of “things” that are distinguishable, understandable, manageable, and can be located with the help of the Internet. In today’s life, almost all things in IoT can beconnectedwith the Internet owing to its communicational and computational capabilities; hence various appropriate and suitable applications can also be realized. Several sensor nodes are collectively deployed for monitoring, sensing, and also for the automation purpose in IoT. The collection of these nodes are generally known as wireless sensor networks (WSNs) and forms an inseparable part of IoT as this technology can sense and monitor any physical things/activities within a particular environment. The aforementioned sensor nodes, also known as “motes,” are cheap, tiny, and are connected internally and distributed in specific areas. These sensor nodes combine multiple features of sensing, computing, and communication through wireless medium and hence in WSNs, physical phenomena are monitored and sensed in real time. Although, WSNs operation is applications specific in terms of the area of interest and way of deployment, but the final aim is monitoring, sensing, broadcasting, and the processing of the collected information. However, the amount of information is huge with an extraordinary rate and that need to be addressed in the current technological world. As known, WSNs are used in a variety of applications, such as military, industry, smart home, healthcare, surveillance, habitat monitoring, and agriculture to name a few. Sensor nodes, thebackbone of WSN, have limited resources, such as energy, computational capability, storage, and communication bandwidth. So, when the demands of WSNs are gradually increases in IoT, more challenges are getting unearthed for the efficient use of it. Moreover, security is another most important concern in WSN-enabled IoT. If an adversary attacks the network and deliberately compromised the nodes, the network security becomes a threat. Therefore, it is required for WSNs to distinguish and eliminate malicious nodes from the network before becoming an active member in the IoT infrastructure.

## LITERATURE SURVEY

#### Analysis of blockchain solutions for IoT: A systematic literature review Authors:

S. K. Loetal

ABSTRACT: The Internet of Things (IoT) aims at connecting things to the Internet in a peer-to- peer paradigm for data collecting and data sharing in our daily life. A blockchain is an immutable append-only ledger maintained by a peer-to-peer network, where the whole network needs to reach a consensus on the transactional data stored on the ledger. With the decentralization nature, the design of IoT and blockchain aligns with each other well. Blockchain has been integrated withthe IoT to solve the existing IoT problems. Our research focuses on analyzing the solutions proposed in academia and the methodologies used to integrate blockchain with the IoT. Through conducting a systematic literature review (SLR) on peer-reviewed, published articles on blockchain-based solutions for IoT, we gather the knowledge on current technical approaches implemented to integrate blockchain into the IoT. Majority of the research in this space is eitherat a conceptual level or at a very early stage. However, we only found 35 published papers with the real implementation of blockchain in the IoT platforms. We elicit the challenges of the IoT that were being addressed, and the detailed design of the blockchain-based solutions from two perspectives, namely data management and thing management. The evaluation methods and metrics used by those works are also being recorded and analyzed. In addition to the analysis of the literature, we provide our insights on improving the existing solutions and research methodology based on our expertise and experience on the blockchain.

#### Computational intelligence in wireless sensor networks: A survey AUTHORS: R. V Kulkarni, S. Member, A. Förster, and G. K. Venayagamoorthy

ABSTRACT: Wireless sensor networks (WSNs) are networks of distributed autonomous devices that can sense or monitor physical or environmental conditions cooperatively. WSNs face many challenges, mainly caused by communication failures, storage and computational constraints and limited power supply. Paradigms of computational intelligence (CI) have been successfully used in recent years to address various challenges such as data aggregation and fusion, energy aware routing, task scheduling, security, optimal deployment and localization. CI provides adaptive mechanisms that exhibit intelligent behavior in complex and dynamic environments like WSNs. CI brings about flexibility, autonomous behavior, and robustness against topology changes, communication failures and scenario changes. However, WSN developers are usually not or not completely aware of the potential CI algorithms offer. On the other side, CI researchers are not familiar with all real problems and subtle requirements of WSNs. This mismatch makes collaboration and development difficult. This paper intends to close this gap and foster collaboration by offering a detailed introduction to WSNs and their properties. An extensive survey of CI applications to various problems in WSNs from various research areas and publication venues is presented in the paper. Besides, a discussion on advantages and disadvantages of CI algorithms over traditional WSN solutions is offered. In addition, a general evaluation of CI algorithms is presented, which will serve as a guide for using CI algorithms for WSNs.

#### IoT Based Wireless Sensor Network for Power Quality Control in Smart Grid AUTHORS: A. H. Bagdadee, M. Z. Hoque, and L. Zhang,

ABSTRACT: The IoT based Power management system requires data from the feeder in the grid. Sufficient power supply with demand is a significant challenge for several countries around the world. Rapid growing demand for power supply requires power quality enhancement to get higher reliability in the smart grid. This smart power system sensor equipped that measures grid power capacity and update to the organization on a consistent schedule. Energy supplies to the specific region indicated by power install capacity in the grid; use a global system for mobile communications (GSM) messaging service to notify customers of power generation and power

supply time. The IoT based wireless Sensor network (WSN) is a revolutionary system for smart monitoring. In this article propose a system demonstrate for the progress and implementation of WSN-based communication systems for smart monitoring and automated control in the electric grid. This work allows for the improvement of grid sharing for maintaining power quality. The dynamic controller has controlled the event of Power quality problem and voltage rise. Appropriate systems and controllers have been demonstrated and analyzed for control performance of a monitoring system in the smart grid.

#### Particle swarm optimization based clustering algorithm with mobile sink for WSNs

AUTHORS: J. Wang, Y. Cao, B. Li, H. J. Kim, and S. Lee

ABSTRACT: Wireless sensor networks with fixed sink node often suffer from hot spots problem since sensor nodes close to the sink usually have more traffic burden to forward during transmission process. Utilizing mobile sink has been shown as an effective technique to enhance the network performance such as energy efficiency, network lifetime, and latency, etc. In this paper, we propose a particle swarm optimization based clustering algorithm with mobile sink for wireless sensor network. In this algorithm, the virtual clustering technique is performed during routing process which makes use of the particle swarm optimization algorithm. The residual energy and position of the nodes are the primary parameters to select cluster head. The control strategy for mobile sink to collect data from cluster head is well designed. Extensive simulation results show that the energy consumption is much reduced, the network lifetime is prolonged, andthe transmission delay is reduced in our proposed routing algorithm than some other popular routing algorithms.

#### Distributed data storage strategy in wireless sensor networks AUTHORS: Z. Song-Juan and Y. Jian,

ABSTRACT: In order to solve the problem of large scalability and low energy efficiency in distributed data storage in wireless sensor networks, the author proposed a temporal-centric storage approach method. By using this method the sensing data are stored in some storage node indexed by a hash function parameterized with detecting time. These nodes act as rendezvous among sink and source nodes. Simulation results show that the proposed approach mitigates the hot-spot problem and can thus improve overall system performance substantially.

## SYSTEM ANALYSIS

#### EXISTING SYSTEM:

However, before discussing the proposed network model and results obtained, recent literatures related to WSNs based IoT with blockchain technology are reviewed briefly for the data storage, authentication and security. The large amount data produced by IoT devices needs to be stored efficiently so that it can be easily retrieved on demand for real time usage. Various challenges during IoT-based data storage in cloud computing have been discussed. Cloud computing based data storage has optimized using hash values which ensure data storage distribution optimization in IoT. Another energy-efficient framework has introduced using fog computing for IoT big data solution in healthcare. The data can be accessible in real time with low latency and delay. Another novel approach has been identified for efficient data management for IoT devices. The performance of the scheme has evaluated in terms of recover-ability and survivability which provide robustness against failure of network within area of interest. Distributed cloud- IoT based solution has involved for optimizing the data among fog nodes/miniclouds within the edge devices. The proposed scheme offers promising results in terms of latency and energy consumption by proper traffic aggregation and processing. The concept of integration of edge computing with sensor nodes has adopted for processing of data locally by compressing the data quickly. The integrated scheme provides effective results which minimize communication overheads by

handling various monitoring, reconfiguration, and data adaption actions. A secure data management and deletion scheme has been introduced using key derivation encryption and data analysis to handle personal information of IoT devices. The sensitive user’s information is encrypted using derivation key algorithm which ensures the privacy of data with reducing the page transfer overheads optimally. Various authentication schemes have been recently developed by different researchers can be seen elsewhere. A mutual authentication, agreement and random node join based smart card authentication for WSNs was developed with particular emphasis on the efficiency of authentication. Another, user efficient authentication method has been introduced without using smart card which provided security against insider attack, theft attack and session recovery attack in any WSNs. Further, to improve the functionality,a three- factor based authentication method has been introduced and that accomplish more privacy and authentication in a particular WSNs. Automated Validation of Internet Security Protocols and Applications (AVISPA) was the next noticeable effort and that utilized formal security verification. Another variation of mutual authentication based scheme used biological information and utilized it with hash and XOR computations which offered sufficient password verification. In the category of user efficient authentication a multi-gateway WSN has been recently developed to accomplish enhanced security. In this exotic approach, the features of most popular schemes, like, password authentication and biometric authenticator are combined to achieve on the desired security. Also, this concept of bio-hashing has been further improved to eradicate the false accept rates without enhancing the false rejection rate efficiently.

Disadvantages: In the existing work, the system cannot resist number of attacks due to post methods used. This system is less performance in which an adversary may deny the contribution of transmitted and received messages or packets to produce confusion for trusted authority.

#### PROPOSED SYSTEM:

The proposed scheme is developed to address security concern using centralized database. Two types of sensor nodes are utilized in the proposed scheme such as regular sensor nodes RSN and cluster

head sensor nodes CHSN. RSN are resource constrained in terms of energy, storage and processing capability. These sensor nodes sense phenomena happen surround and forward the gathered information to CHSN. CHSN is responsible for gathering information from RSN and forward information to Base station act as a Trusted Authority BTA. BTA is responsible for certification of all sensor nodes. Initially, the legitimacy of sensor nodes is granted by BTA before joining the network. Sensor nodes will get the authentication information and different parameters from BTA. Further, the sensor RSN forwards sensed information to CHSN. Further, the information is forwarded by CHSN towards BTA through wireless medium, therefore it is very easy for attackers to stole and forge the data such as location, speed, identity and sensed information during transmission. Hence, block chain based privacy-preserving scheme is proposed to mitigate such problems.  The proposed scheme is completed into various steps such as initialization phase, Registration phase, sensor node authentication phase, message signing and verification phase, key update phase and revocation phase and tracing phase. Initially, all the parameters required for all phases are computed by BTA. After that, all regular sensor nodes can initialize process by providing their information (like location, speed, identity, residual energy and sensed information) to CHSN. Further, CHSN broadcasts all information including its own information towards BTA. After collecting the information from CHSN, BTA utilizes that information to construct an untamperable Key Mechanism (UKM) and then allocates the UKM to all CHSN. Then, CHSN stored UKM and further keys are distributed among regular senor nodes.

Advantages: A block chain-based solution for privacy preserving and authentication with cloud storage, Base station provides certification to all sensor nodes, Certification key of all nodes are stored in an Untamperable Key Mechanism. Large amount of sensed information are stored in clouds.

## SYSTEM REQUIREMENTS:

This section elaborates on the functional requirements of the application. The SRS itself can be divided into module, each module having specifications. In order to carry out the project, the following hardware and software is required.

#### HARDWARE REQUIREMENTS:

* System: i3
* Hard Disk: 40 GB
* Floppy Drive: 1.44 Mb
* Monitor: 15 VGA Color
* Mouse: Logitech
* Ram: 512 Mb

#### SOFTWARE REQUIREMENTS:

* Technology: Java 2 Standard Edition, JDBC
* Web Server: Tomcat 7.0
* Client Side Technologies: HTML, CSS, JavaScript
* Server Side Technologies: Servlets , JSP
* Data Base Server: MySQL
* Editor: Netbeans8.1

## SYSTEM STUDY

#### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

#### ECONOMICAL FEASIBILITY

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

#### TECHNICAL FEASIBILITY

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

#### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# SYSTEM DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed ofseveral steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

### SOFTWARE DESIGN

In designing the software following principles are followed:

1. Modularity and partitioning: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.
2. Coupling: modules should have little dependence on other modules of a system.
3. Cohesion: modules should carry out in a single processing function.
4. Shared use: avoid duplication by allowing a single module be called by other that need the function it provides
   1. SYSTEM ARCHITECTURE

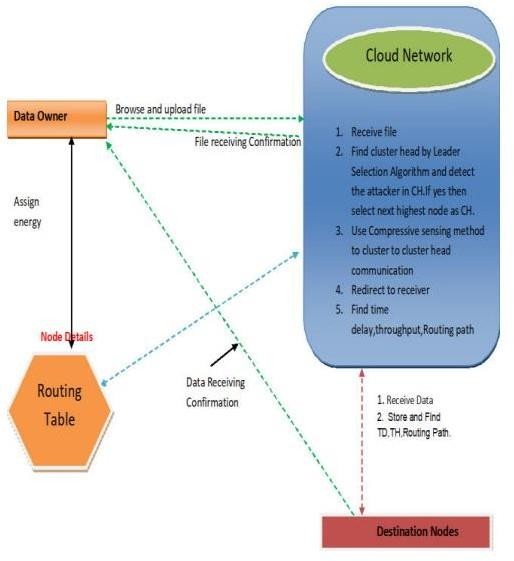


Fig 2.1 System Architecture

## UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### GOALS:

The Primary goals in the design of the UML are as follows:

* Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
* Provide extendibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development process.
* Provide a formal basis for understanding the modeling language.
* Encourage the growth of OO tools market.
* Support higher level development concepts such as collaborations, frameworks, patterns and components.
* Integrate best practices.

## Use case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

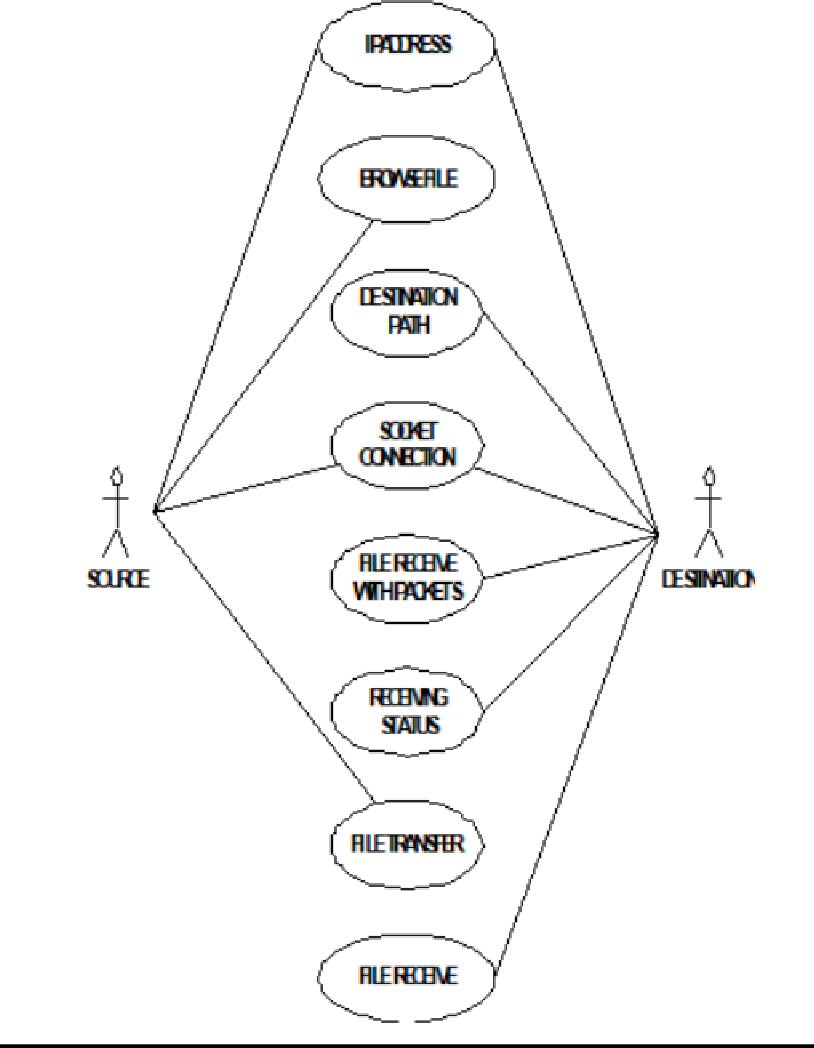


Fig 2.2.1 Use Cse Diagram

## Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



Fig 2.2.2 Class Diagram

## Object diagram:

The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

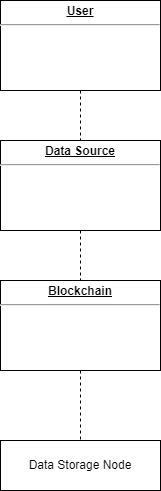


Fig 2.2.3 Object Diagram

## State diagram:

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.



Fig 2.2.4 State Diagram

## Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

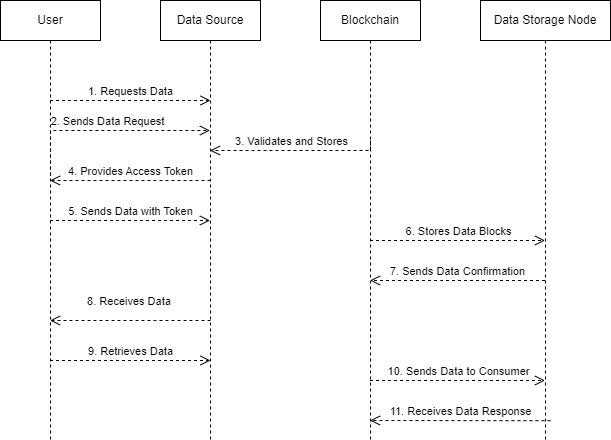


Fig 2.2.5 Sequence Diagram

# 3. MODULE DESIGN

The major modules of the project are

Data Owner: In this module, the Data Owner will browse the data file and then send to the particular Nodes. Data Owner will send their data file to router and router will connect to clusters, in a cluster highest energy sensor node will be activated and send to particular Nodes (A, B, C…). And if any attacker will change the energy of the particular sensor node, then Data Owner will reassign the energy for sensor node. Cloud Network the Cloud Network manages a multiple clusters (cluster1, cluster2, cluster3, and cluster4) to provide data storage service. In cluster n-number of nodes (n1, n2, n3, n4…) are present, and in a cluster the sensor node which have more energy considered as a cluster head and it will communicate first. In a router Data Owner can view the node details, view routing path, view time delay and view attackers. Router will accept the file from the Data Owner, the cluster head will select first andit size will reduced according to the file size, then next time when we send the file, the other node will be cluster head. Similarly, the cluster head will select different node based on highest energy. The time delaywill be calculated based on the routing delay.

Cluster as Block Chain In cluster n-number nodes are present and the clusters are communicates with every clusters (cluster1, cluster2, cluster3 and cluster4).In a cluster the sensor node which has more energy considered as a cluster head. The Data Owner will assign the energy for each & every node. The Data Owner will upload the data file to the router; in a router clusters are activated and the cluster based networks, to select the highest energy sensor nodes, and send to particular Nodes.

Nodes (End User) in this module, the Nodes can receive the data file from the Data Owner via router. The Nodes receive the file by without changing the File Contents. Users may receive particular data files within the network only.

Attacker is one who is injecting the fake energy to the corresponding sensor nodes. The attacker decries the energy to the particular sensor node. After attacking the nodes, energy will be changed in a router.

## INPUT/OUTPUT DESIGN

Input design: Considering the requirements, procedures to collect the necessary input data in most efficiently designed. The input design has been done keeping in view that, the interaction of the user with the system being the most effective and simplified way.

Also the measures are taken for the following

* + - Controlling the amount of input
    - Avoid unauthorized access to the classroom.
    - Eliminating extra steps
    - Keeping the process simple
    - At this stage the input forms and screens are designed.

Output design: All the screens of the system are designed with a view to provide the user with easy operations in simpler and efficient way, minimum key strokes possible. Instructions and important information is emphasized on the screen. Almost every screen is provided with no error and important messages and option selection facilitates. Emphasis is given for speedy processing and speedy transaction between the screens. Each screen assigned to make it as much user friendly as possible by usinginteractive procedures. So to say user can operate the system without much help from the operating manual.

## SOFTWARE DEVELOPMENT TOOLS

#### HTML

Html is a language which is used to create web pages with html marking up a page to indicate its format, telling the web browser where you want a new line to begin or how you want text or images aligned and more are possible.

We used the following tags in our project.

#### Table

Tables are so popular with web page authors is that they let you arrange the elements of a web page in such a way that the browser won’t rearrange them web page authors frequently use tables to structure webpages.

#### TR

TR is used to create a row in a table encloses <TH> and <TD> elements. <TR> contain many attributes. Some of them are,

* ALIGN: specifies the horizontal alignment of the text in the table row.
* BGCOLOR: Specifies the background color for the row.
* BORDERCOLOR: Sets the external border color for the row.
* VALIGN: Sets the vertical alignment of the data in this row.

#### TH:

TH is used to create table heading.

* ALIGN: Sets the horizontal alignment of the content in the table cell. Sets LEFT, RIGHT, CENTER.
* BACKGROUND: Species the back ground image for the table cell.
* BGCOLOR: Specifies the background color of the table cell
* VALIGN: Sets the vertical alignment of the data. Sets to TOP, MIDDLE, BOTTOM or BASELINE.
* WIDTH: Specifies the width of the cell. Set to a pixel width or a percentage of the display area.

#### TD:

TD is used to create table data that appears in the cells of a table.

* ALIGN: Species the horizontal alignment of content in the table cell. Sets to LEFT, CENTER, RIGHT.
* BGCOLOR: Specifies the background image for the table cell.
* BGCOLOR: sets the background color of the table cells.
* WIDTH: Species the width of the cell

#### Frames:

Frames are used for either run off the page or display only small slices of what are supposed to be shown and to configure the frame we can use <FRAMESET>There are two important points to consider when working with <FRAMESET>.

* <FRAMESET> element actually takes the place of the <BODY> element in a document.
* Specifying actual pixel dimensions for frames .

<FRAME> Elements are used to create actual frames.

From the frameset point of view dividing the browser into tow vertical frames means creating two columns using the <FRAMESET> elements COLS attribute.

The syntax for vertical fragmentation is,

<FRAMESET COLS =”50%, 50%”>

</FRAMESET>

Similarly if we replace COLS with ROWS then we get horizontal fragmentation. The syntax for horizontal fragmentation is,

<FRAMESET ROWS=”50%, 50%”>

</FRAMESET>

#### Form:

The purpose of FORM is to create an HTML form; used to enclose HTML controls, like buttons and text fields.

#### Attribute:

* ACTION: Gives the URL that will handle the form data.
* NAME: Gives the name to the form so you can reference it in code set to an alphanumeric string.
* METHOD: method or protocol is used to sending data to the target action URL. The GET method is the default, it is used to send all form name/value pair information in an URL. Using the POST method, the content of the form are encoded as with the GET method, but are sent in environment variables.

#### Controls in HTML

<INPUT TYPE =BUTTON>

It creates an html button in a form. ATTRIBUTES

* NAME: gives the element a name. Set to alphanumeric characters.
* SIZE: sets the size.
* VALUE: sets the caption of the element.

<INPUT TYPE = PASSWORD>

Creates a password text field, which makes typed input. ATTRIBUTES

* NAME: gives the element a name, set to alphanumeric characters.
* VALUE: sets the default content of the element.

<INPUT TYPE=RADIO>

Creates a radio button in a form. ATTRIBUTE

* NAME: Gives the element a name. Set to alphanumeric character.
* VALUE: Sets the default content of the element.

<INPUT TYPE=SUBMIT>

Creates a submit button that the user can click to send data in the form back to the web server. ATTRIBUTES

* NAME: Gives the element a name. Set to alphanumeric characters.
* VALUE: Gives this button another label besides the default, Submit Query. Set to alphanumeric characters.

<INPUT TYPE=TEXT>

Creates a text field that the user can enter or edit text in. ATTRIBUTES

* NAME: Gives the element a name. Set to alphanumeric characters.
* VALUE: Holds the initial text in the text field. Set to alphanumeric characters.

#### Java Script:

Java script originally supported by Netscape navigator is the most popular web scripting language today. Java script lets you embedded programs right in your web pages and run these programs using the web browser. You place these programs in a <SCRIPT> element, usually within the <HEAD> element. If you want the script to write directly to the web page, place it in the <BODY> element.

#### Java script Methods:

Writeln

Document.writeln () is a method, which is used to write some text to the current web page. onClick

Occurs when an element is clicked. onLoad

Occurs when the page loads. onMouseDown

Occurs when a mouse button goes down. onMouseMove

Occurs when the mouse moves. OnUnload

Occurs when a page is unloaded.

#### MySQL:

MySQL is an open source relational database management system (RDBMS).This is the most popular database system used with PHP. MySQL is distributed and supported by Oracle Corporation.

MySQL runs on almost all platforms including Linux, Unix and Windows. Although it can be used in a wide range of applications, MySQL is often associated with web applications and online publishing.

MySQL is an essential constituent of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as an operating system, in the form of Apache web server, MySQL relational database management system and PHP object-oriented scripting language.

#### Advantages of MySQL:

Data Security: MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications including WordPress, Drupal, Joomla, Facebook and Twitter.

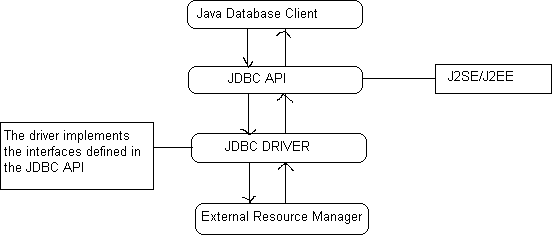
High Performance: MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance.

Round-the-Clock Up-time: MySQL comes with the assurance of 24×7 up-time and offers a wide range of high-availability solutions, including specialized cluster servers and master/slave replication configurations.

The Flexibility of Open Source: All the fears and worries that arise in an open-source solution can be brought to an end with MySQL’s round-the-clock support and enterprise indemnification. The secure processing and trusted software of MySQL combine to provide effective transactions for large-volume projects. It makes maintenance, debugging and upgrades fast and easy while enhancing the end-user experience.

#### JDBC Drivers

The JDBC API only defines interfaces for objects used for performing various database-related tasks like opening and closing connections, executing SQL commands, and retrieving the results. We all write our programs to interfaces and not implementations. Either the resource manager vendor or a third party provides the implementation classes for the standard JDBC interfaces. These software implementations are called JDBC drivers. JDBC drivers transform the standard JDBC calls to the external resource manager- specific API calls. The diagram below depicts how a database client written in java accesses an external resource manager using the JDBC API



Depending on the mechanism of implementation, JDBC drivers are broadly classified into four types.

#### TYPE 1

Type 1 JDBC drivers implement the JDBC API on top of a lower level API like ODBC. These drivers are not generally portable because of the independency on native libraries. These drivers translate the JDBC

calls to ODBC calls and ODBC sends the request to external data source using native library calls. The JDBC-ODBC driver that comes with the software distribution for J2SE is an example of a type1 driver.

#### TYPE 2

Type 2 drivers are written in mixture of java and native code. Type 2 drivers use vendors specific native APIs for accessing the data source. These drivers transform the JDBC calls to vendor specific calls using the vendor’s native library. These drivers are also not portable like type1 drivers because of the dependency on native code.

#### TYPE 3

Type 3 drivers use an intermediate middleware server for accessing the external data sources. The calls to the middleware server are database independent. However, the middleware server makes vendor specific native calls for accessing the data source. In this case, the driver is purely written in java.

#### TYPE 4

Type 4 drivers are written in pure java and implement the JDBC interfaces and translate the JDBC specific calls to vendor specific access calls. They implement the data transfer and network protocol for the target resource manager. Most of the leading database vendors provide type4 drivers for accessing their database servers.

#### Driver manager and Driver:

The java.sql package defines an interface called Java.sql.Driver that makes to be implemented by all the JDBC drivers and a class called java.sql.DriverManager that acts as the interface to the database clients for performing tasks like connecting to external resource managers, and setting log streams. When a JDBC client requests the DriverManager to make a connection to an external resource manager, it delegates the task to an appropriate driver class implemented by the JDBC driver provided either by the resource manager vendor or a third party.

#### JAVA.SQL.DRIVERMANAGER

The primary task of the class driver manager is to manage the various JDBC drivers register. It also provides methods for:

* Getting connections to the databases.
* Managing JDBC logs.
* Setting login timeout.

#### Managing drivers

JDBC clients specify the JDBC URL when they request a connection. The driver manager can find a driver that matches the request URL from the list of register drivers and delegate the connection requestto that driver if it finds a match JDBC URLs normally take the following format:

#### <protocol>:<sub-protocol>:<resource>

The protocol is always jdbc and the sub-protocol and resource depend on the type of resource manager. The URL for postgreSQL is in the format:

#### Jdbc: postgres ://< host> :< port>/<database>

Here host is the host address on which post master is running and database is the name of the database to which the client wishes to connect.

#### Managing controls

DriverManager class is responsible for managing connections to the databases:

public static Connection getConnection (String url,Properties info) throws SQLException

This method gets a connection to the database by the specified JDBC URL using the specified username and password. This method throws an instance of SQLException if a database access error occurs.

#### Connetions

The interface java.sql.Connection defines the methods required for a persistent connection to the database. The JDBC driver vendor implements this interface. A database ‘vendor-neutral’ client never uses the implementation class and will always use only the interface. This interface defines methods for the following tasks:

* Statements, prepared statements, and callable statements are the different types of statements for issuing sql statements to the database by the JDBC clients.
* For getting and setting auto-commit mode.
* Getting meta information about the database.
* Committing and rolling back transactions.

#### Creating connections

The interface java.sql.Connection defines a set of methods for creating database statements. Database statements are used for sending SQL statements to the database:

Public Statement createStatement () throws SQLException

This method is used for creating instances of the interface java.sql.Statement. This interface can be used for sending SQL statements to the database. The interface java.sql.Statement is normally used for sending SQL statements that don’t take any arguments. This method throws an instance of SQLException if a database access error occurs:

Public Statement createStatement (int resType, int resConcurrency) throws SQLException.

#### JDBC result set:

A JDBC result set represents a two dimensional array of data produced as a result of executing SQL SELECT statements against databases using JDBC statements. JDBC result sets are represented by the interface java.sql.ResultSet. The JDBC vendor provider provides the implementation class for this interface.

#### Scrolling resultset

public booleannext() throws SQLException public booleanprevious() throws SQLException public booleanfirst() throws SQLException

public booleanlast() throws SQLException

#### Statement

The interface java.sql.stament is normally used for sending SQL statements that do not have IN or OUT parameters. The JDBC driver vendor provides the implementation class for this interface. The common methods required by the different JDBC statements are defined in this interface. The methods defined by java.sql. Statement can be broadly categorized as follows:

* Executing SQL statements
* Querying results and resultsets
* Handling SQL batches
* Other miscellaneous methods

The interface java.sql.statements defines methods for executing different SQL statements like SELECT, UPDATE, INSERT, DELETE, and CREATE.

Public Resultset execute Query (string sql) throws SQLException

The following figure shows how the DriverManager, Driver, Connection, Statement, ResultSet classes are connected.

DriverManager

Driver

Application

Connection

Driver

Result Set

Statement

Prepared Statement

Callable Statement

Result Set Result Set

#### JAVA SERVER PAGES (JSP)

Java Server Pages (JSP) technology enables you to mix regular, static HTML with dynamically generated content. You simply write the regular HTML in the normal manner, using familiar Web-page-building tools. You then enclose the code for the dynamic parts in special tags, most of which start with <% and end with %>

#### The need of JSP:

Servlets are indeed useful, and JSP by no means makes them obsolete. However, It is hard to write and maintain the HTML.

* You cannot use standard HTML tools.
* The HTML is inaccessible to non-Java developers.

#### Benefits of JSP

JSP provides the following benefits over servlets alone:

It is easier to write and maintain the HTML In this no extra backslashes, no double quotes, and no lurking Java syntax.

* You can use standard Web-site development tools: We use Macromedia Dreamweaver for most of the JSP pages. Even HTML tools that know nothing about JSP can used because they simply ignore the JSP tags.
* You can divide up your development team: The Java programmers can work on the dynamic code. The Web developers can concatenate on the representation layer. On large projects, this division is very important. Depending on the size of your team and the complexity of your project, you can enforce a weaker or stronger separation between the static HTML and the dynamic content.

#### Creating template text

A large percentage of our JSP document consists of static text known as template text. In almost all respects, this HTML looks just likes normal HTML follows all the same syntax rules, and simply “passed through” to that client by the servlet created to handle the page. Not only does the HTML look normal, it can be created by whatever tools you already are using for building Web pages.

There are two minor exceptions to the “template text passed through” rule. First, if you want to have <%0r

%> in the out port, you need to put <\% or %\> in the template text. Second, if you want a common to appear in the JSP page but not in the resultant document,

<%-- JSP Comment -- %> HTML comments of the form:

<!—HTML Comment -->

are passed through to the client normally.

#### Types of JSP scrolling elements:

JSP scripting elements allow you to insert Java code into the servlet that will be generated from the JSP page. There are three forms:

* Expressions of the form <%=Java Expression %>, which are evaluated and inserted into the servlet’s output.
* Sciptletsof the form <%Java code %>, which are inserted into the servlet’s\_jspService method (called by service).
* Declarations of the form<%! Field/Method Declaration %>, which are inserted into the body of the servlet class, outside any existing methods.

#### Using JSP Expressions

A JSP element is used to insert values directly into the output. It has the following form:

<%= Java Expression %>

The expression is evaluated, converted to a string, and inserted in the page. This evaluation is performed at runtime (when the page is requested) and thus has full access to the information about the request For example, the following shows the date/time that the page was requested.

Current time: <%=new java.util.Date () %>

#### Predefined variables

To simplify expressions we can use a number of predefined variables (or “implicit objects”). The specialty of these variables is that, the system simple tells what names it will use for the local variables in

\_jspService. The most important ones of these are:

* request, the HttpServletRequest.
* response, the HttpServletResponse.
* session, the HttpSession associated with the request
* out, the writer used to send output to clients.
* application, the ServletContext. This is a data structure shared by all servlets and JSP pages in the web application and is good for storing shared data.

Here is an example:

Your hostname: <%= request.getRemoteHost () %>

#### Comparing servlets to JSP pages:

JSP works best when the structure of the HTML page is fixed but the values at various places need to be computed dynamically. If the structure of the page is dynamic, JSP is less beneficial. Some times servlets are better in such a case. If the page consists of binary data or has little static content, servlets are clearly superior. Sometimes the answer is neither servlets nor JSP alone, but rather a combination of both.

#### Writing scriptlets:

If you want to do something more complex than output the value of a simple expression .JSP scriptlets let you insert arbitrary code into the servlet’s jspService method. Scriptlets have the following form:

<% Java code %>

Scriptlets have access to the same automatically defined variables as do expressions (request, response, session, out , etc ) .So for example you want to explicitly send output of the resultant page , you could use the out variable , as in the following example:

<% String queryData = request.getQueryString (); out.println (“Attached GET data: “+ queryData) %>

#### Scriptlet Examples:

As an example of code that is too complex for a JSP expression alone, a JSP page that uses the bgColor request parameter to set the background color of the page .Simply using

<BODY BGCOLOR=”<%= request.getParameter (“bgcolor”) %> “> would violate the cardinal rule of reading form data.

#### Using declarations

A JSP declaration lets you define methods or fields that get inserted into the main body of the servlet class .A declaration has the following form:

<%! Field or Method Definition %>

Since declarations do not generate output, they are normally used in conjunction with JSP expressions or scriptlet. In principle, JSP declarations can contain field (instance variable) definitions, method definitions, inner class definitions, or even static initializer blocks: anything that is legal to put inside a class definition but outside any existing methods. In practice declarations almost always contain field or method definitions.

We should not use JSP declarations to override the standard servlet life cycle methods. The servlet into which the JSP page gets translated already makes use of these methods. There is no need for declarations to gain access to service, doget, or dopost, since calls to service are automatically dispatched to \_jspService , which is where code resulting from expressions and scriptlets is put. However for initialization and cleanup, we can use jspInit and jspDestroy- the standard init and destroy methods are guaranteed to call these methods in the servlets that come from JSP.

#### Jakarta Tomcat

Tomcat is the Servlet/JSP container. Tomcat implements the Servlet 2.4 and JavaServer Pages 2.0 specification. It also includes many additional features that make it a useful platform for developing and deploying web applications and web services.

#### Terminology:

Context – a Context is a web application.

$CATALINA\_HOME – This represents the root of Tomcat installation.

#### Directions and files

/bin – Startup, shutdown, and other scripts. The \*.sh files (for Unix systems) are functional duplicates of the \*.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.

/conf – Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.

/logs – Log files are here by default.

/webapps – This is where webapps go\

#### Installation

Tomcat will operate under any Java Development Kit (JDK) environment that provides a JDK 1.2 (also known as Java2 Standard Edition, or J2SE) or later platform. JDK is needed so that servlets, other classes, and JSP pages can be compiled.

#### Deployment directions for default web applications

HTML and JSP Files

* Main Location

$CATALINA\_HOME/webapps/ROOT

* Corresponding URLs. <http://host/SomeFile.html> <http://host/SomeFile.jsp>
* More Specific Location (Arbitrary Subdirectory).

$CATALINA\_HOME/webapps/ROOT/SomeDirectory

* Corresponding URLs <http://host/SomeDirectory/SomeFile.html> <http://host/SomeDirectory/SomeFile.jsp>

#### Individual Servlet and Utility Class Files

* Main Location (Classes without Packages).

$CATALINA\_HOME/webapps/ROOT/WEB-INF/classes

* Corresponding URL (Servlets). <http://host/servlet/ServletName>
* More Specific Location (Classes in Packages).

$CATALINA\_HOME/webapps/ROOT/WEB-INF/classes/packageName

* Corresponding URL (Servlets in Packages). <http://host/servlet/packageName.ServletName>

#### Servlet and Utility Class Files Bundled in JAR Files

* Location

$CATALINA\_HOME/webapps/ROOT/WEB-INF/lib

* Corresponding URLs (Servlets) <http://host/servlet/ServletName> <http://host/servlet/packageName.ServletName>

#### XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P).

It is simply a web server if we want to make a website or designing and make a working website then XAMPP is useful .it gives an environment of hiw server works.

1. It contains apache, mysql, filezilla servers by which we can use them and helps us in login and logout sessions, cookies we gives a good help in websites
2. Also, it has word press feature by which it contains many themes of websites which are popular and we can use them to make a website without using so much php coding ,HTML, CSS etc.
3. How to use it: 1. if we are working on mysql then we just on the server of mysql and go to php admin page.
4. To work on php based web pages we just on the server and then, code on a notepad by using php pages.

# SOFTWARE TESTING

## What do you mean by software testing?

Testing involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they shouldn'tor things don't happen when they should. It is oriented to 'detection'.

## Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. This testing mode is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

Unit tests are written from a programmer's perspective. They ensure that a particular method of a class successfully performs a set of specific tasks. Each test confirms that a method produces the expected output when given a known input.

## Performance Testing

Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative attributes such as

Reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

Performance testing can verify that a system meets the specifications claimed by its manufacturer or vendor. The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

Performance testing can also be used as a diagnostic aid in locating communications bottlenecks. Often a system will work much better if a problem is resolved at a single point or in a single component. For example, even the fastest computer will function poorly on today's Web if the connection occurs at only 40 to 50 Kbps (kilobits per second).

## Integration Testing

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit isdefined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integrationtesting is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

## Test case 1

Test case for Login form:

|  |  |
| --- | --- |
| **FUNCTION:** | **LOGIN** |
| **EXPECTED RESULTS:** | Should Validate the user and check his existence in database |
| **ACTUAL RESULTS:** | Validate the user and checking the user against the database |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

## Test case 2:

Test case for User Registration form:

|  |  |
| --- | --- |
| **FUNCTION:** | **USER REGISTRATION** |
| **EXPECTED RESULTS:** | Should check if all the fields are filled by the user and saving the user to database. |
| **ACTUAL RESULTS:** | Checking whether all the fields are field by user or not through validations and saving user. |
| **LOW PRIORITY** | **No** |

|  |  |
| --- | --- |
| **HIGH PRIORITY** | **Yes** |

## Test case 3

#### Test case for Change Password

When the old password does not match with the new password, then this results in displaying an error message as “OLD PASSWORD DOES NOT MATCH WITH THE NEW PASSWORD”.

|  |  |
| --- | --- |
| **FUNCTION:** | **Change Password** |
| **EXPECTED RESULTS:** | Should check if old password and new password fields are filled by the user and saving the user to database. |
| **ACTUAL RESULTS:** | Checking whether all the fields are field by user or not through validations and saving user. |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

## Test case 4:

#### Test case for Forget Password

When a user forgets his password he is asked to enter Login name, ZIP code, Mobile number. If these are matched with the already stored ones then user will get his Original password.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modu le** | **Function ality** | **Test Case** | **Expected Results** | **Actual Results** | **Result** | **Priority** |
| User | Login Usecase | 1. Navigate To Www.Sample.Com 2. Click On Submit Button Without Entering Username and Password | A Validation Should Be As Below “Please Enter Valid Username & Password” | A  Validation Has Been Populated As Expected | Pass | High |
|  |  | 1. Navigate To Www.Sample.Com 2. Click On Submit Button With Out Filling Password And With Valid Username | A Validation Should Be As Below “Please Enter Valid Password Or Password Field Can Not Be Empty’ | A  Validation Is Shown As Expected | Pass | High |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 1. Navigate To [Www.Sample.Com](http://www.sample.com/) 2. Enter Both Username And Password Wrong   And Hit Enter | A Validation Shown As Below “The Username Entered Is Wrong” | A  Validation Is Shown As Expected | Pass | High |
|  |  | 1. Navigate To Www.Sample.Com 2. Enter Validate Username And Password And Click On Submit | Validate Username And Password In Database And Once if they are correct then show the main page | Main Page Home Page Has Been Displayed | Pass | High |

Fig 4.7 Test Case Table

# Experimental Results

## Cloud Network

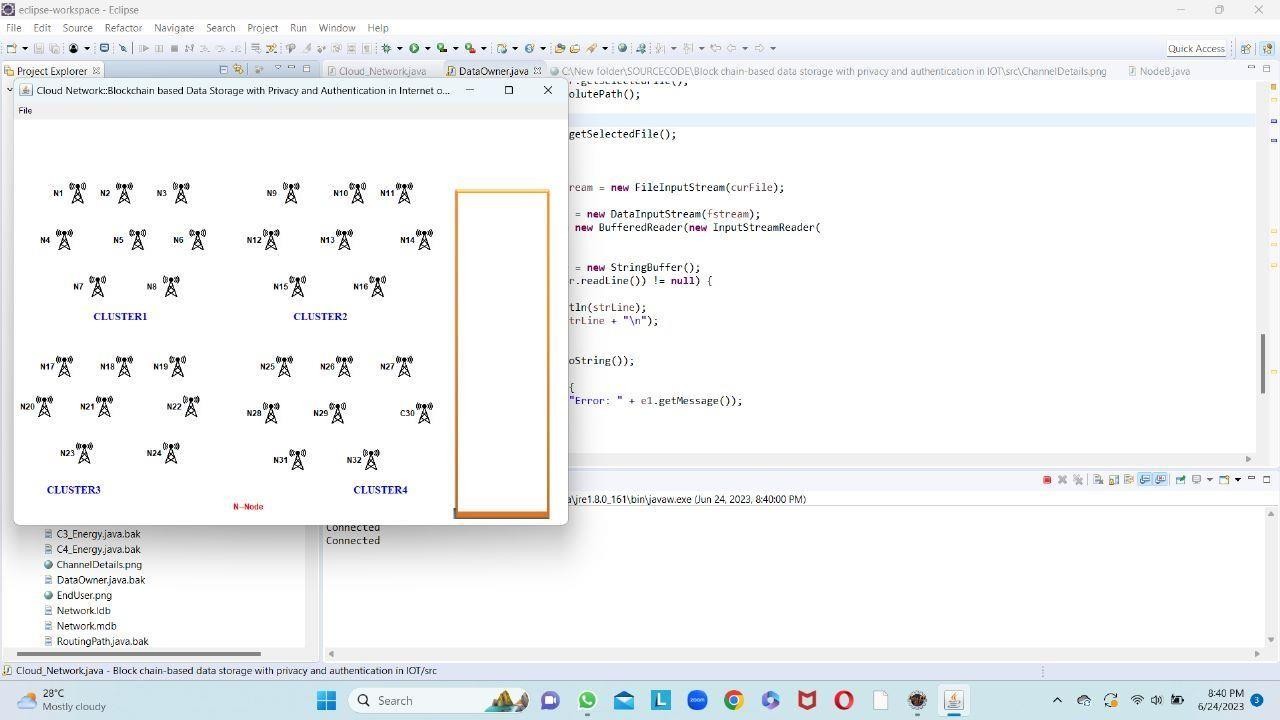


Fig 5.1 Cloud Network

## Base Station

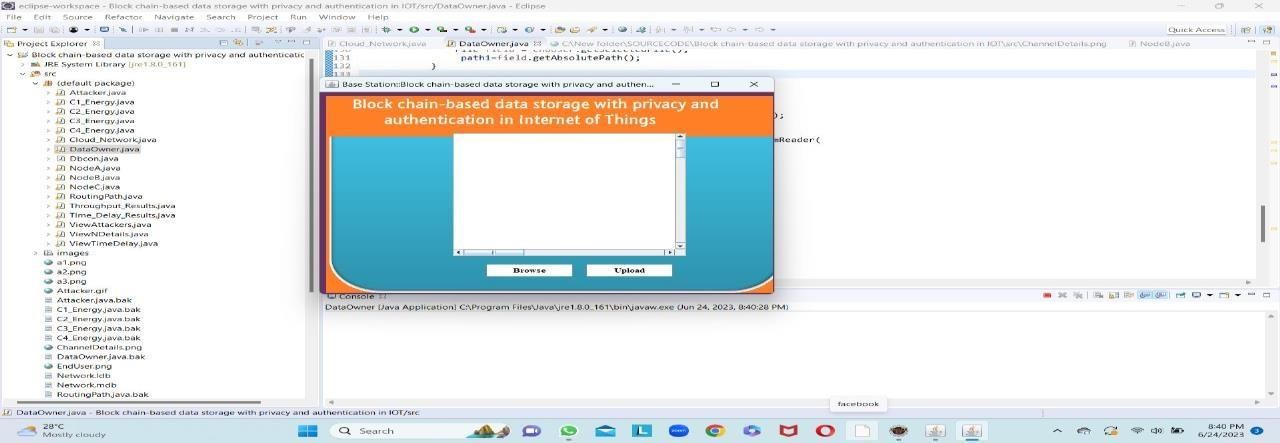


Fig 5.2 Base Station

## Receiver

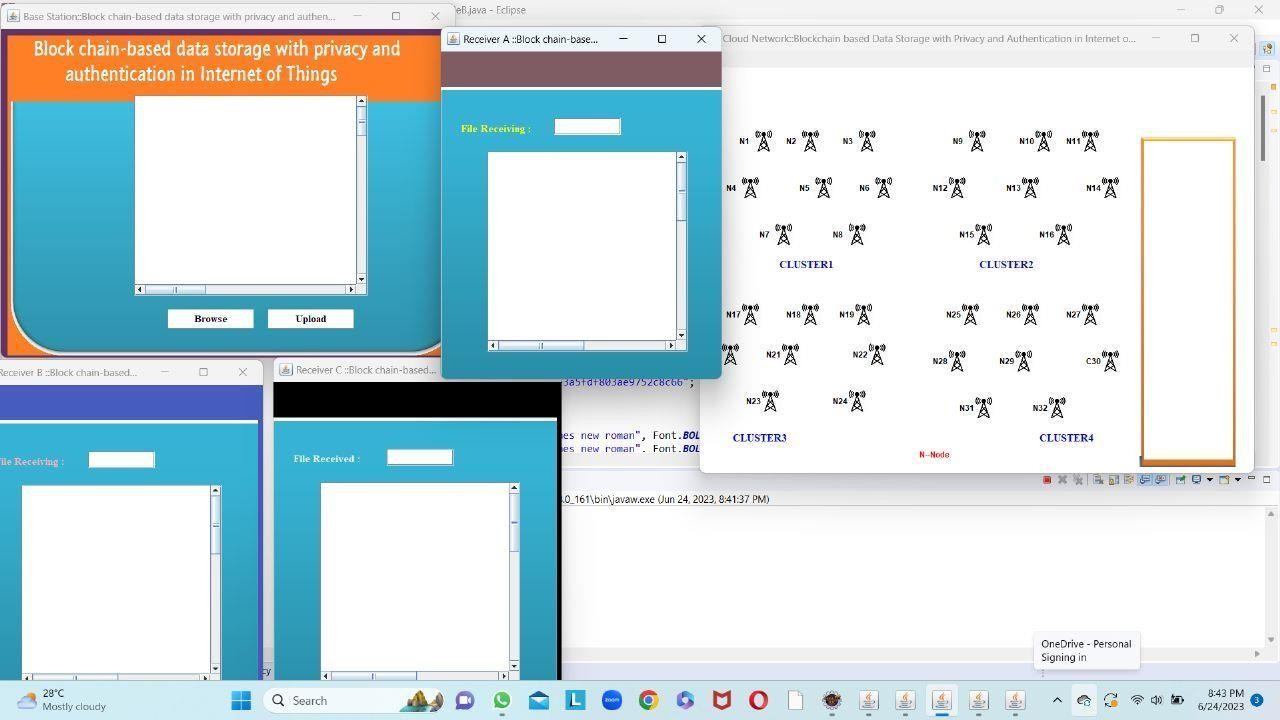


Fig .5.3 Receiver

## Selection of file

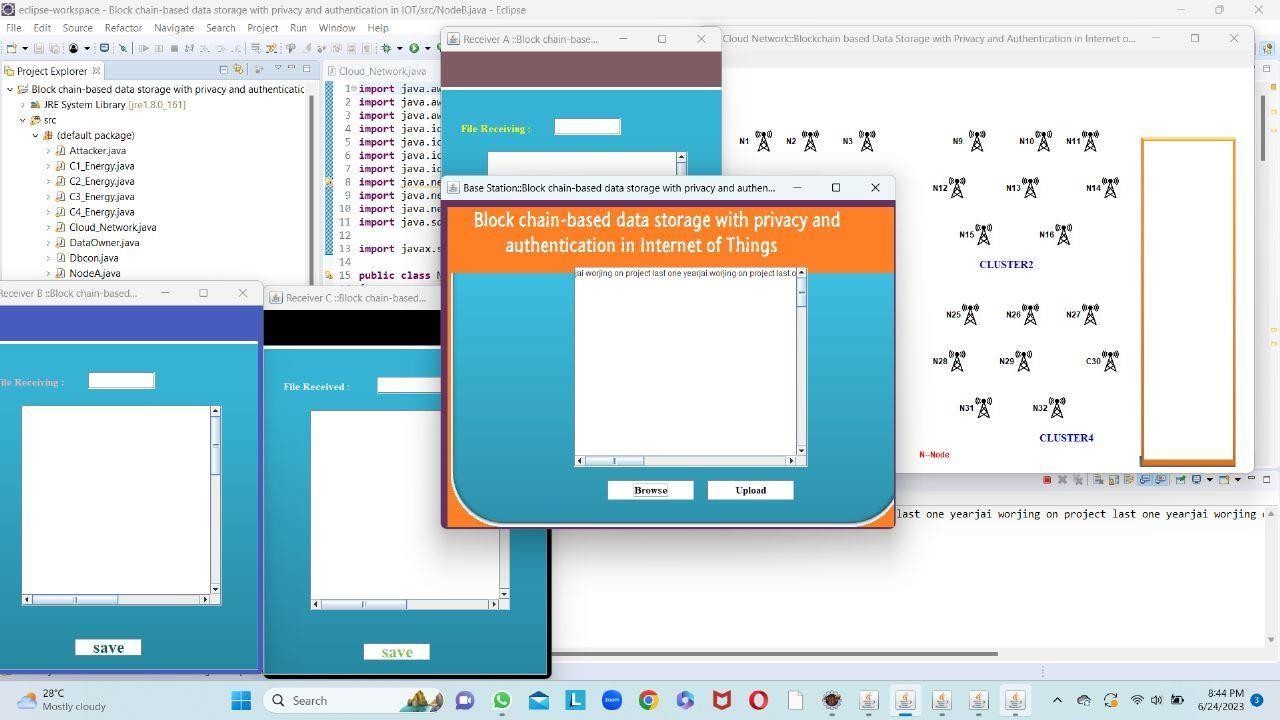


Fig 5.4 Selection of File

## Transferring of File in Block Chain through Clusters

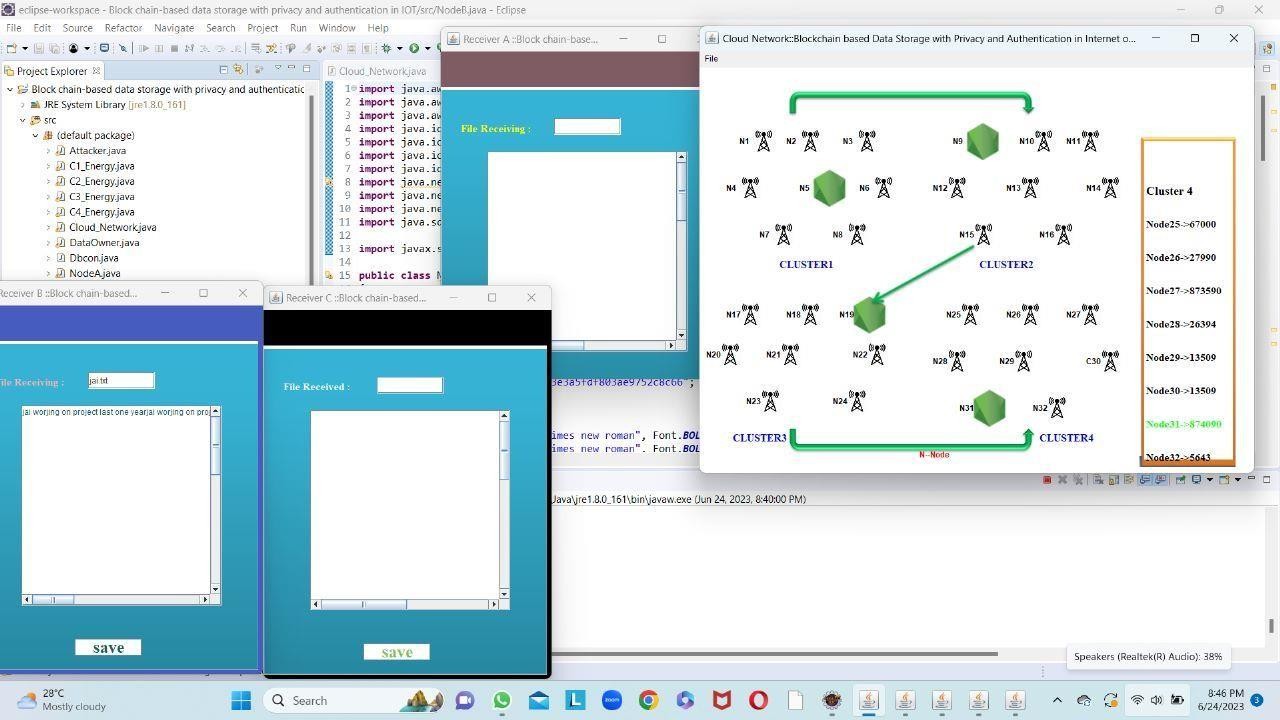
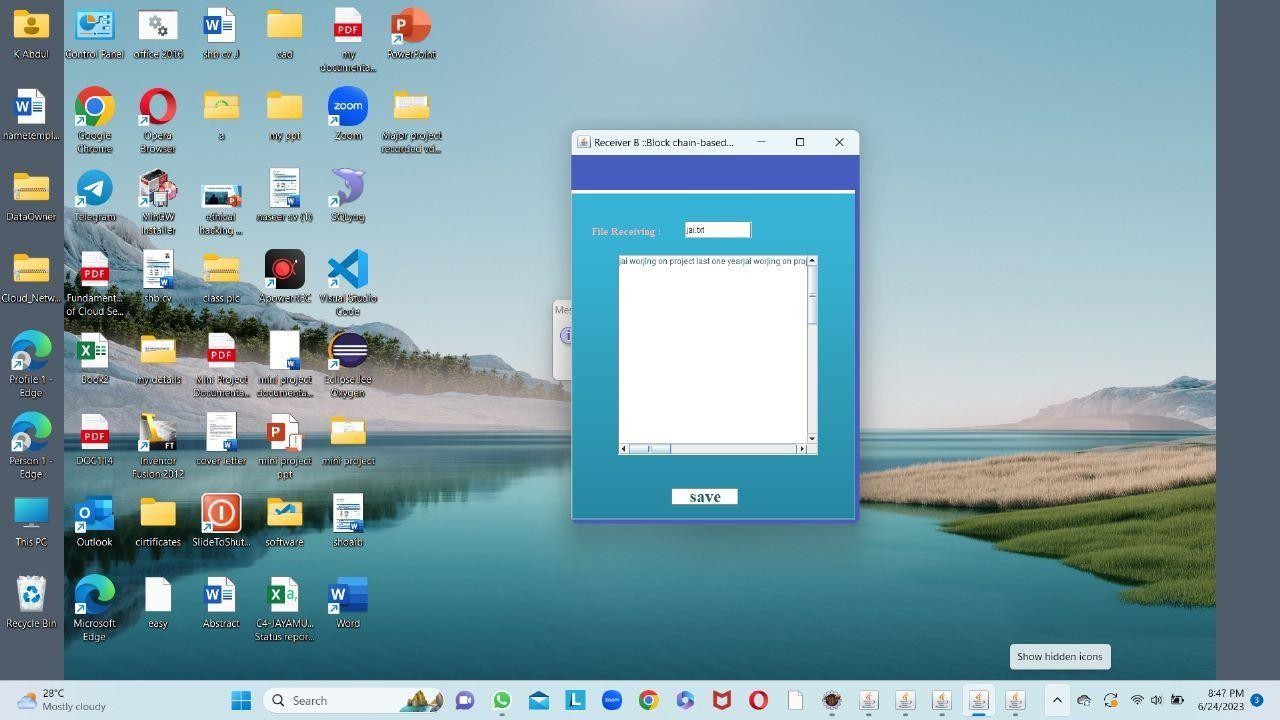


Fig 5.5 Transferring of File in Block Chain through Clusters

## File Received at the Destination Receiver



5.6 File Received at the Destination Receiver

# CONCLUSION AND FUTURE SCOPE

A privacy-preserving authentication scheme based on blockchain with cloud data storage was accomplished effectively for the WSN-enabled IoTs. Initially, the process of registration and certification for all sensor nodes was performed by BS. After completing the certification process, all the keyparameters were stored in UKM controlled by the cluster heads. Further, the cluster heads broadcast the collected information from its members to BS and the information is then separated into two parts: 1) key parameters and 2) sensed information. The large amount of these sensed data was then shared with cloud for more reliable and efficient storage. The key parameters were further recorded on emerging blockchaintechnology to improve the immutability and transparency of the obtained data. The certification revocation process successfully eliminated malfunctioning sensor nodes. The proposed scheme accomplished better results in terms of detection accuracy, certification delay, and computational overheads. The simulated results and comparative analysis demonstrate that the proposed algorithm achieves 19.33% better results in terms of average of detection accuracy. Sharing large amount of information into cloud storage ensured reliability and effectiveness of the proposed scheme. In the future, we shall try to optimize the data management and resources of the framework for effective results.

## Future Scope

The future scope of blockchain-based data storage with privacy and authentication in the Internet of Things (IoT) holds immense potential for innovation and advancement. Here are some possible areas of future development:

* + - Scalability: One significant challenge with blockchain technology is scalability. Future research and development efforts may focus on enhancing the scalability of blockchain networks to handle the increasing volume of data generated by IoT devices. Implementing techniques like sharding, side chains, or layer-2 solutions could help improve scalability and transaction throughput.
    - Interoperability: As the IoT ecosystem continues to expand, achieving interoperability between different blockchain networks and IoT devices becomes crucial. Future advancements may aim to standardize protocols and develop interoperability frameworks to ensure seamless data exchange and collaboration across diverse IoT devices and blockchain platforms.
    - Enhanced Privacy and Security: Privacy and security are paramount in IoT systems. Future developments could focus on strengthening privacy-preserving techniques, such as zero- knowledge proofs, homomorphic encryption, and secure multi-party computation. Additionally, advancements in identity management and authentication mechanisms can enhance the security of IoT devices and the blockchain network.
    - Smart Contracts and Automation: Smart contracts enable self-executing agreements and automate processes in blockchain networks. Future advancements may explore the integration of IoT devices with smart contracts to automate data verification, access control, and data sharing. This can streamline operations, reduce costs, and enable trustless interactions between IoT devices.
    - Integration with AI and Machine Learning: AI and machine learning technologies can bring intelligence to IoT systems. Future developments may leverage the combination of blockchain, IoT, and AI to enhance data analytics, predictive maintenance, and anomaly detection. This integration can enable more intelligent decision-making and optimize IoT operations.
    - Energy Efficiency: Blockchain networks, particularly those using Proof of Work (PoW) consensus mechanisms, consume significant energy. Future research may focus on developing energy- efficient consensus algorithms, such as Proof of Stake (PoS), and exploring alternative energy sources to minimize the environmental impact of blockchain-based IoT systems.
    - Regulatory and Legal Frameworks: The widespread adoption of blockchain-based IoT systems necessitates the development of robust regulatory and legal frameworks. Governments and regulatory bodies may work towards defining standards, data governance policies, and privacy regulations specific to blockchain-based IoT deployments to ensure compliance, protect user rights, and address legal challenges.
    - Real-World Implementations and Industry Adoption: As the technology matures, there will likely be increased real-world implementations and industry adoption of blockchain-based data storage solutions in the IoT domain. Organizations may leverage the benefits of decentralized, secure, and transparent data storage to enhance their IoT deployments, improve data integrity, and enablenew business models.

# BIBLIOGRAPHY

1. Y. A. Abdulrahman, M. Kamalrudin, S. Sidek, and M. A. Hassan, “Internet of Things: Issues and challenges,” J. Theor. Appl. Inf. Technol., vol. 94, no. 1, pp. 52–60, 2016.
2. S. K. Lo et al., “Analysis of blockchain solutions for IoT: A systematic literature review,” IEEE Access, vol. 7, pp. 58822–58835, 2019.
3. R. V Kulkarni, S. Member, A. Förster, and G. K. Venayagamoorthy, “Computational intelligence in wireless sensor networks: A survey,” IEEE Commun. Surveys Tuts., vol. 13, no. 1, pp. 68–96, 1st Quart., 2011.
4. A. H. Bagdadee, M. Z. Hoque, and L. Zhang, “IoT based wireless sensor network for power quality control in smart grid,” Procedia Comput. Sci., vol. 167, pp. 1148–1160, 2020. [Online]. Available: doi.org/10.1016/j.procs.2020.03.417
5. J. Wang, Y. Cao, B. Li, H. J. Kim, and S. Lee, “Particle swarm optimization based clustering algorithm with mobile sink for WSNs,” Future Gener. Comput. Syst., vol. 76, pp. 452–457, Nov. 2017.
6. Z. Song-Juan and Y. Jian, “Distributed data storage strategy in wireless sensor networks,” Int. J. Online Eng., vol. 12, no. 11, pp. 52–57, 2016.
7. L. Buttyan, D. Gessner, A. Hessler, and P. Langendoerfer, “Application of wireless sensor networks in critical infrastructure protection: Challenges and design options security and privacy in emerging wireless networks,” IEEE Wireless Commun., vol. 17, no. 5, pp. 44–49, Oct. 2010.
8. R. Singh, D. K. Singh, and L. Kumar, “A review on security issues in wireless sensor network,” J. Inf. Syst. Commun. vol. 2, no. 7, pp. 28–34, 2010.
9. H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “IoT-based big data storage systems in cloud computing: Perspectives and challenges,” IEEE Internet Things J., vol. 4, no. 1, pp. 75–87, Feb. 2017.
10. M. Wang and Q. Zhang, “Optimized data storage algorithm of IoT based on cloud computing in distributed system,” Comput. Commun., vol. 157, pp. 124–131, May 2020.
11. C. Feng, M. Adnan, A. Ahmad, A. Ullah, and H. U. Khan, “Towards energy-efficient framework for IoT big data healthcare solutions,” Sci. Program., vol. 2020, pp. 1–9, May 2020.
12. M. Asiri, T. Sheltami, L. Al-Awami, and A. Yasar, “A novel approach for efficient management of data lifespan of IoT devices,” IEEE Internet Things J., vol. 7, no. 5, pp. 4566–4574, May 2020.
13. P. Maiti, J. Shukla, B. Sahoo, and A. K. Turuk, “Efficient data collection for IoT services in edge computing environment,” in Proc. Int. Conf. Inf. Technol. (ICIT), 2018, pp. 101–106.
14. M. A. Serhani, H. T. El-Kassabi, K. Shuaib, A. N. Navaz, B. Benatallah, and A. Beheshti, “Self- adapting cloud services orchestration for fulfilling intensive sensory data-driven IoT workflows,” Future Gener. Comput. Syst., vol. 108, pp. 583–597, Jul. 2020.
15. J. Xiong et al., “A secure data deletion scheme for IoT devices through key derivation encryption and data analysis,” Future Gener. Comput. Syst., vol. 111, pp. 741–753, Oct. 2020.
16. T. H. Chen and W. K. Shih, “A robust mutual authentication protocol for wireless sensor networks,” ETRI J., vol. 32, no. 5, pp. 704–712, 2010.
17. N. Bruce, Y. J. Kang, H. R. Kim, S. H. Park, and H. J. Lee, A Security Protocol Based-on Mutual Authentication Application Toward Wireless Sensor Network (Lecture Notes in Electrical Engineering), vol. 339. Heidelberg, Germany: Springer, 2015, pp. 27–34.
18. K. Chatterjee, A. De, and D. Gupta, “A secure and efficient authentication protocol in wireless sensor network,” Wireless Pers. Commun., vol. 81, no. 1, pp. 17–37, 2015.
19. A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, “A dynamic password-based user authentication scheme for hierarchical wireless sensor networks,” J. Netw. Comput. Appl., vol. 35, no. 5, pp. 1646– 1656, 2012.
20. T. Maitra, R. Amin, D. Giri, and P. D. Srivastava, “An effcient and robust user authentication scheme for hierarchical wireless sensor networks without tamper-proof smart card,” Int. J. Netw. Security, vol. 18, no. 3, pp. 553–564, 2016.