**Blockchain Technology for Secure Supply Chain Management: A Comprehensive Review**

**ABSTRACT**

Supply chain management (SCM) is a core corporate activity responsible for moving commodities and services from one point to another through a variety of stakeholders. The traditional SCM is based on a centralized approach managed at the central headquarter, and all other sub-offices get instructions from the main office. Some major issues with present SCM systems are security, transactional transparency, traceability, stakeholder involvement, product counterfeiting, additional delays, fraud, and instabilities. Blockchain (BC) emerges as a technology that can manage the data and build trust efficiently and transparently. It can also aid in transaction authorization and verification in the supply chain or payments without a third party. To address the present SCM issues, BC technology is a feasible solution. Motivated by the aforementioned considerations, in this paper, we present a survey on the adoption of BC in SCM. This paper undertakes a comprehensive analysis of the literature on BC characteristics, implementations, and business consequences in various SCM. This Blockchain-centered study, in particular, discloses the research state and delineates future research directions by studying and analyzing 97 up-to-date publications highlighting BC’s supply chain uses. Transparency and traceability, information sharing, product anti-counterfeiting, and building trust are the major aspects propelling BC’s implementation in SCM. Further, we analyzed various applications of SCM in which BC can be used as a probable technology to secure all transactions. Then, we have highlighted open issues and research challenges for adopting BC technology in SCM that open the doors for beginners eager to start work in this amazing area.

**LIST OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| 1. | **CHAPTER 1: INTRODUCTION**   * 1. General   2. Scope Of the Project   3. Objective   4. Problem Statement   5. Existing System   1.5.1 Existing System Disadvantages  1.5.2 Literature Survey  1.6 Proposed System  1.6.1 Proposed System Advantages |  |
| 2. | **CHAPTER 2: PROJECT DESCRIPTION**   * 1. General   2.2 Methodologies  2.2.1 Modules Name  2.2.2 Modules Explanation  2.2.3 Given Input and Expected Output  2.3 Technique or Algorithm |  |
| 3. | **CHAPTER 3: REQUIREMENTS**  3.1 General  3.2 Hardware Requirements  3.3 Software Requirements  3.4 Functional Specification  3.5 Non-Functional Specification |  |
| 4. | **CHAPTER 4: SYSTEM DESIGN**  4.1 General  4.1.1 Use Case Diagram  4.1.2 Class Diagram  4.1.3 Object Diagram  4.1.4 State Diagram  4.1.5 Sequenced Diagram  4.1.6 Collaboration Diagram  4.1.7 Activity Diagram  4.1.8 Component Diagram  4.1.9 E-R Diagram  4.1.01 Data Flow Diagram  4.1.11 Deployment Diagram  4.2 System Architecture |  |
| 5. | **CHAPTER 5: SOFTWARE SPECIFICATION**  5.1 General  5.2 Features of Java  5.2.1 The Java Framework  5.2.2 Objective of Java  5.2.3 Java Swing Overview  5.2.4 Evolution of Collection Framework  5.3 Conclusion |  |
| 6. | **CHAPTER 6: IMPLEMENTATION**  6.1 General  6.2 Implementation |  |
| 7. | **CHAPTER 7: SNAPSHOTS**  7.1 General  7.2 Various Snapshots |  |
| 8. | **CHAPTER 8: SOFTWARE TESTING**  8.1 General  8.2 Developing Methodologies  8.3 Types of Testing  8.3.1 Unit testing  8.3.2 Functional test  8.3.3 System Test  8.3.4 Performance Test  8.3.5 Integration Testing  8.3.6 Acceptance Testing  8.3.7 Build the test plan |  |
| 9. | **CHAPTER 9: APPLICATIONS AND FUTURE ENHANCEMENT**  9.1 General  9.2 Future Enhancements |  |
| 10. | **CHAPTER 10: CONCLUSION & REFERENCES**  10.1 Conclusion  10.2 References |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 2.3.2 | Module Diagram |  |
| 4.2 | Use case Diagram |  |
| 4.3 | Class Diagram |  |
| 4.4 | Object Diagram |  |
| 4.5 | State Diagram |  |
| 4.6 | Sequence Diagram |  |
| 4.7 | Collaboration Diagram |  |
| 4.8 | Activity Diagram |  |
| 4.9 | Component Diagram |  |
| 4.10 | Data flow Diagram |  |
| 4.11 | E-R Diagram |  |
| 4.12 | Deployment Diagram |  |
| 4.13 | System Architecture |  |
| 7.1 | Home Page |  |

**LIST OF NOTATIONS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **NAME** | **NOTATION** | | **DESCRIPTION** | |
| 1. | Class | *Class Name*  *-attribute-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | | Represents a collection of similar entities grouped together. | |
| 2. | Association | name  Class B  Class A  Class A  Class B | | Associations represents static relationships between classes. Roles represents the way the two classes see each other. | |
| 3. | Aor | Class A  Class A  Class B  Class B | | It aggregates several classes into a single classis | |
| 5. | Aggregation | Interaction between the system and external environment | |
| 5. | Relation  (uses) | | Uses | | Used for additional process communication. | |
| 6. | Relation  (extends) | | extends | | Extends relationship is used when one use case is similar to another use case but does a bit more. | |
| 7. | Communication | |  | | Communication between various use cases. | |
| 8. | State | | State | | State of the process. | |
| 9. | Initial State | |  | | Initial state of the object | |
| 10. | Final state | |  | | Final state of the object | |
| 11. | Control flow | |  | | Represents various control flow between the states. | |
| 12. | Decision box | |  | | Represents decision making process from a constraint | |
| 13. | Usecase | |  | | Interact ion between the system and external environment. | |
| 14. | Component | |  | | Represents physical modules which is a collection of components. | |
| 15. | Node | |  | | Represents physical modules which are a collection of components. | |
| 16. | Data Process / State | |  | | A circle in DFD represents a state or process which has been triggered due to some event or action. | |
| 17. | External entity | |  | | Represents external entities such as keyboard, sensors, etc. | |
| 18. | Transition | |  | | Represents communication that occurs between processes. | |
| 19. | Object Lifeline | |  | | Represents the vertical dimensions that the object communications. | |
| 20. | Message | | Message | | Represents the message exchanged. | |

**LIST OF ABBREVATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| 1**.** | DB | Data Base |
| 2. | JVM | Java Virtual Machine |
| 3. | JSP | Java Server Page |
| 4. | CB | Collective Behavior |
| 5. | RSSS | Ramp secret sharing scheme |
| 6. | JRE | Java Runtime Environment |

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL**

SCM is defined as the movement of goods from producer to consumer. It is a network that is made up of independent or semi-independent business entities such as producers, suppliers, retailers, and clients who are involved in the manufacturing and distribution of goods. It covers all from item improvement, sourcing, generation, coordination, and the data frameworks required to facilitate these exercises. Since early times, supply chains have occurred, starting with the first item shaped or service formed and sold. But with industrialization and globalization, SCM got to be more refined, permitting companies to do a more proficient work of creating and conveying merchandise and administrations. Organizations can now identify failure before it occurs and take proactive measures to prevent it. They make an exact estimate that supports meeting client requests and monetary goals simultaneously. Every node in the supply chain network must be concurred upon and flexible in response to the needs of the consumer and competent in dealing with issues such as tracking, exchange approaches, shipping modes, and so on. Consumers also have various options for purchasing products, including in-store, online, and more, putting them in the driver’s seat when it comes to defining SCM priorities. So, while discussing the primary issues in today’s SCM, we discover that

1. Today’s supply chain is centralized. However, a centralized supply chain is often time-consuming and expensive to manage, and it lacks essential functionality for market analysis.
2. The complexity of supply chains and value networks adds a high cost to supply chain players, which the customer eventually absorbs. The data in the majority of the documents are kept on BC, which is an expensive procedure.
3. The present supply chain architecture does not deliver the requisite level of transparency and traceability.
4. In the current supply chain, stakeholders and consumers face dilemmas relating to coordination, inventory management, human resource reliance, order management, stock management, expiry date, etc. As a result, stakeholders cannot evaluate demand and hence are unable to optimize output and storage.

The traditional SCM is based on a centralized approach. A single business headquarters and a single warehouse full of departmental managers in different areas like logistics, distribution, and procurement, and these managers are responsible for overseeing their specific location during the complete supply chain. They keep track of the information in a centralized database stored nearby. When the data on the record is not beneficial to the company’s growth, it may be misrepresented secretly. As a result, mistrust between ventures has progressively noticeable, resulting in higher communication expenses. Also, there is no pricing transparency in the supply chain because of the middlemen. Furthermore, because of the high risk of data manipulation inside the venture, the data across supply chain entities is incompatible; as a result, the product tracing procedure has been delayed.

**1.2 SCOPE OF THE PROJECT**

This Blockchain-centered study, in particular, discloses the research state and delineates future research directions by studying and analyzing 97 up-to-date publications highlighting BC’s supply chain uses. Transparency and traceability, information sharing, product anti-counterfeiting, and building trust are the major aspects propelling BC’s implementation in SCM. Further, we analyzed various applications of SCM in which BC can be used as a probable technology to secure all transactions. Then, we have highlighted open issues and research challenges for adopting BC technology in SCM that open the doors for beginners eager to start work in this amazing area. BC solutions need to be evaluated in terms of their scalability and cost-effectiveness to address the concerns of business managers.

**1.3 OBJECTIVE**

This paper investigates the current status of BC technology implementation in various supply chain network areas. We comprehensively surveyed the application of BC in Food and Health supply chain networks. This study makes several significant contributions, including theoretical advances related to the adoption of BC in the supply chain.

Following are the research contributions of this paper:

* To explore the opportunities and benefits of using BC to enhance various functions in different supply chain networks.
* To present various areas of application in the supply chain where BC can be applied.
* To explore and present various challenges in implementing BC technology in the present and future supply chain networks.

**1.4 PROBLEM STATEMENT**

In Today’s supply chain, there is no encrypted mechanism to store consumers’ private information. Cyber-attacks will be able to access this data, revealing important public and personal information. Another key issue is that goods only travel in one direction in today’s supply chain management. As a result, if a product is faulty, the customer is responsible for the consequences. He didn’t have an option but to accept the chance. As a result, in traditional SCM, enabling the reverse flow of products and transactions for each customer is a major challenge. Supply chain attacks are another key issue in Today’s SCM. Instead of directly attacking a single organization, supply chain attacks target vendors and providers. It is a cyber-attack that targets a reputable third-party vendor who provides critical services or software to the SC. Unsecure suppliers in a chain are attacked to obtain access to their bigger trade partners in a supply chain assault.

* 1. **EXISTING SYSTEM**
* Most existing tracking and traceability system, which is used by most supply chain networks, has problems with centralized management and data privacy.
* BC, the fundamental technology, has gotten a lot of interest from academics and business in the last several years due to its nature of immutability and decentralization.
* The adoption of BC technology has answered the requirement for a safe, transparent, and traceable platform. Most existing tracking and traceability system, which is used by most supply chain networks, has problems with centralized management and data privacy.
  + 1. **EXISTING SYSTEM DISADVANTAGES**
* The existing operating system and how much time it would take to develop a new BC-based system.
* The existing system unable to avoid the server energy cost.

**1.5.2 LITERATURE SURVEY**

**Title:** Blockchain technology and trust in supply chain management: A literature review and research agenda

**Year:** 2021

**Author:** A. Batwa and A. Norrman.

**Description:** Applying blockchain technology for information sharing in supply chain is driven by many factors, but developing trust is one of the most proposed. However, trust is a multidimensional, intangible concept without an agreed-upon definition. Whereas some argue that trust is the main driver of blockchain technology, others have found a negative relationship. This study focuses on how applying blockchain in supply chain management can influence trust and proposes a corresponding research agenda. Trust and blockchain technology discussions are scattered throughout the literature. Thus, a systematic literature review was performed based on a conceptual trust framework. This study discovered a gap in linking trust theories to blockchain technology applications especially in supply chain management, and provided insights into trust’s reciprocal nature. Current literature strongly expects trust as a consequence for blockchain adoption if considered in the technology. Simultaneously, trust in supply chain partners is strongly expected as an antecedent to blockchain because it requires openness in information sharing. Thus, propositions and agenda for future research are suggested. The research is limited to literature findings due to the immaturity and low scalability of blockchain technology adoption; however, the most reviewed articles in less than two years old, increase the results’ accuracy.

**Title:** Asurvey paper on blockchain technologies in supply chain management

**Year:** 2021

**Author:** R. Raman, J. Sushmitha, and M. K. Nalini.

**Description:**

The supply chain connects organisations, resources, activities, people, and information in order to transform natural resources and raw materials into a finished product for delivery to the ultimate client. The complexity of supply chains and value networks has increased over time. These complexities add a huge cost to the participants of supply chain, which is ultimately borne by the customer. The existing infrastructure also fails to provide desired amount of transparency and traceability into the supply chain. The blockchain technology is a revolutionary concept that can overcome the problems in the existing infrastructure to build trust, provide transparency and traceability across the supply chain value network. This paper is a survey of various blockchain based solutions that have been proposed to be implemented in various fields of supply chain management. This paper also briefly discusses the application of this technology in various sectors. The growth of industries has led to the discovery of a management technique called supply chain management. Supply chain basically refers to the complete network of people, organizations, resources, technologies, resources and the activities, from the start stage to the final end-product stage. The term supply chain management refers to the management of flow of all the entities who are a part of the supply chain. This covers the flow of goods, services, information, and money between two supply chain members, as well as the movement and storage of raw materials, work-in-process inventories, and finished product flow from point of production to point of consumption, as well as end-to-end order fulfilment. The demand for a safe, transparent, and traceable platform has been answered by the usage of Blockchain technology as more sectors aim to improve the efficiency of their supply chain management processes.

**Title:** Industrial blockchain: A state-of-the-art survey

**Year:** 2021

**Author:** Z. Li, R. Y. Zhong, Z.-G. Tian, H.-N. Dai, A. V. Barenji, and G. Q. Huang.

**Description:** As an underlying and backbone technology of Bitcoin, Blockchain attracted extensive attention worldwide in recent years due to its unique characteristics of decentralization, openness, immutability, anonymity, etc., which enables it to build a trust basis through recording the point-to-point decentralized transactions in an immutable way via the attached timestamp, thereby improving system efficiency and reducing the cost without relying on the central agent. As it is considered to be a potentially revolutionary technology, Blockchain has been introduced into various industrial fields including finance, supply chain, manufacturing, healthcare, energy, and smart city. In this paper, we conduct a state-of-the-art survey of industrial Blockchain in terms of published articles between 2017 and 2020, and worldwide Blockchain movement including North America, Europe, and the Asia Pacific region so far. We conduct a statistic analysis of the collected articles in terms of three dimensions, which are year of publication, leading research institutes and researchers, and article classification to present a multi-dimensional trend or conclusion. Besides, we analyse articles that are cited over a certain number of times in detail to investigate the hot research directions. Finally, the challenges, opportunities, and future perspectives are discussed to summarize the main obstacles of industrial Blockchain and identify the open research questions in the near future.

Blockchain technology has grabbed global attention with the prosperity of Bitcoin, which was originally proposed by Satoshi Nakamoto in 2008. It is a new application paradigm of multiple computer technologies including asymmetric encryption, distributed network, peer-to-peer transmission, smart contract, and consensus mechanism, etc. to create permanent, immutable, authorized and time-stamped records of transactions, which enables it to establish mutual trust at low costs in an untrusted competitive environment without any third parties. Although Blockchain was designed as the underlying technology of cryptocurrency originally, the application potential of Blockchain is already far beyond finance.

**Title:** Analysing the impact of blockchain technology for operations and supply chain management: An explanatory model drawn from multiple case studies.

**Year:** 2020

**Author:** S. Tonnissen and F. Teuteberg.

**Description:** Blockchain technology is said to have a high disruptive potential and can do without an intermediary. Numerous contributions deal with its impact on and possibilities for logistics and supply chains. In this article, we use a multiple case analysis to develop an explanatory model for the interaction of actors in an operational supply chain involving blockchain technology. In addition, we show which intermediary tasks the blockchain could replace and what impact this would have on the industry logic. For this purpose, we analyze the status quo in practice based on a multiple case study with real use cases and find answers to our research questions. The findings of the paper include (1) insights into the impact of blockchain technology on the logistics industry, and (2) the implications and research questions related to blockchain technology and the impact of blockchain technology on business models.

The blockchain is a distributed system between participants in a network which stores transactions between those participants in a consistent, unchanging and chronological chain. Due to the peer-to-peer network of the blockchain, there are no intermediaries between the players; thus, in a logistics chain, producers or suppliers can deal directly with their customers. Business relationships between unknown individuals are necessarily based on trust and trust is an inherent component of the blockchain consensus mechanism. Disintermediation or the bypassing of middlemen promises to achieve supply chain management goals of cost, quality, speed, reliability, risk reduction, sustainability, and flexibility. However, it is still unclear what effect blockchain technology will have on roles and tasks in a logistics chain and which intermediary tasks the blockchain will take on. Players in a logistics chain face new players with new functions as well as a loss of business partners that they know well.

**Title:** Blockchain implementations and use cases for supply chains-A survey

**Year:** 2020

**Author:** P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni.

**Description:** Since Bitcoin's debut in 2008, blockchain, the technology behind the cryptocurrency, has been gaining increasing scientific and industrial interest. Due to the technology's innate distributed and immutable features, the adoption of blockchains on supply chains is one of the most promising recent applications. In this survey, we review academic researches and implementations of distributed ledgers on supply chains. We present the current state of research on the subject and summarize the benefits and the challenges of the distributed organization and management of supply chains. Focusing on industrial practices and use cases, we discuss the technical characteristics and maturity of the various industrial projects. Our goal is to assess the applicability of blockchains in the supply chain domain and to provide a foundation for practitioners and researchers to direct their future projects towards improving the technology and its applications.

Distributed ledger technologies found themselves in the spotlight after the publication of Nakamoto’s white paper on Bitcoin [1], the cryptocurrency that uses Blockchain, the most popular distributed ledger to this day. Ever since, thousands of scientific papers, blog articles, industry guides and financial reports have been written on what is a blockchain and the ways in which it has entered the industrial, financial and technological worlds.

**Title:** Dairy supply chain system based on blockchain technology

**Year:** 2020

**Author:** S. Shingh, V. Kamalvanshi, S. Ghimire, and S. Basyal.

**Description:** With the increase in the complexity of the dairy supply chain system, consumers know very less about the products produced or processed by producer or processor. Such information asymmetry present in the dairy industry has serious concern over human health, environmental sustainability, and welfare issues. In this context, we require the effective dairy supply chain system that not only fulfils the information requirement of the consumers but also increase the trust of the consumer on the dairy product they are consuming. The paper tries to present the application of Blockchain technology in the dairy sector. It focuses on the application of the Blockchain technology on improving the dairy supply chain system. This paper presents how this technology can be used in the dairy supply chain system and outlines the potential benefit of it to the different stakeholders and the whole dairy industry as a whole. This is a review article based on the secondary data and information that are obtained from various published articles. Keywords: Blockchain technology; internet of things; agriculture; dairy supply chain system; food.

Due to the presence of the various types of nutrients like carbohydrate, proteins, fat, minerals, vitamins, and other various essential components in milk, it is often claimed as the perfect food. However, in the present time due to the malpractices involved during the processing, handling and transportation of the milk, consumption of the milk available in the market can have deadly harm in human health. In the last decades, there is an increase in the occurrences of milk-related scandals all over the globe. Milk related incidences are higher in the developing countries like China, India, and Pakistan. In China, milk scandal of 2008 has left more than 3 million victims and the death of six children. A study done by the Food Safety and Standard Authority of India (FSSAI) shows that 68.4% of the milk in the country is not as per the legal standard Milk related incidences are also high in developed countries like the USA. In the year 2018, there was a recall of 160 products in the USA due to undeclared allergens, in which one third was due to milk. Such incidents have increased the concern for the traceability of the milk and milk products. However, concern for dairy traceability is not limited to public health safety. Other issues include environmental sustainability, animal and farm labour welfare.

* 1. **PROPOSED SYSTEM**
* This Blockchain-centered study, in particular, discloses the research state and delineates future research directions by studying and analyzing 97 up-to-date publications highlighting BC's supply chain uses. Transparency and traceability, information sharing, product anticounterfeiting, and building trust are the major aspects propelling BC's implementation in SCM.
* The potential of this technology to track all types of transaction more transparently and securely motivate us to explore the possibilities BC offers across the supply chain.
* Data handling with monitoring and regulating data in a virtual environment, less paperwork, increased efficiency with faster response times and increased supply chain visibility of adopting BC in supply chains.
  + 1. **PROPOSED SYSTEM ADVANTAGES**
* Cost effectiveness will be provided.
* BC solutions need to be evaluated in terms of their scalability and cost-effectiveness to address the concerns of business managers.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL**

BC solutions need to be evaluated in terms of their scalability and cost-effectiveness to address the concerns of business managers. This is a serious challenge, and a big question is what would happen to the existing operating system and how much time it would take to develop a new BC-based system. In this criterion, search using BC Technology, Supply Chain Management, BC Use in SCM, Provenance, and Supply Chain Traceability. Finally, we examined the abstracts and conclusions of the collected publications and determined the relevant techniques and ideas for the proposed survey.

**2.2 METHODOLOGIES**

**2.2.1** **MODULES NAME:**

**This project having the following 5 modules:**

**1. User Interface Design**

**2. Management Team**

**3. OEM**

**4. Design Engineer**

**5. Manufacturing Manager**

**6. Dealer**

**2.2.2** **MODULES EXPLANATION AND DIAGRAM**

* **User Interface Design**

In this module we design the windows for the project. In this module mainly we are focusing the login design page with the Partial knowledge information. Application Users need to view the application they need to login through the User Interface GUI is the media to connect User and Media Database and login screen where user can input his/her user name, password and password will check in database, if that will be a valid username and password then he/she can access the database.

User Login

User Registration

Login verification

User Page

Error Page

User

Cloud Database

**2. Management Team**

This is the first module of this project. In this module Management Team will register, and then MT will login. Management will add the Design Engineers, Manufacturing Manager, and MRO. And view the repair details in any manufacturing system. Management team will verify the production details and until where production completed. They will get orders from dealer, and then will send the orders to different locations. They will get the financial reports and future budget information.

MT

Add DE, MM, and MRO

View Production Details

Get Orders and Deliver them

Repair Details

Database

**3. OEM**

This is the second module of this project. In this module OEM will register, and then login. OEM will add the OEM Persons to different locations. Then OEM person will see the requirements from Design Engineer, and then OEM person will send the required equipment. And then see the status of that equipment.

OEM

OEM person will send Equipment

See the Specifications

Add Person

Download Specifications

Verify the Status of Equipment

Database

**4. Design Engineer**

This is the third module of the project. In this module Design Engineer will login. Then DE will send specifications to OEM Person. Then DE will verify those equipment’s with the specifications. Then if those are perfect then DE will approve otherwise DE will reject those. If accepted then DE will forward to Manufacturing Manager.

Design Engineer

Verify the Equipments

Get the Equipment

Send Specifications

Accept or Reject the Equipment

Forward to Manufacturing Manager

Database

**5. Manufacturing Manager**

This is the fourth module of this project. In this module MM will login. Then MM will add the Manufacturing persons, then forward work to them. And MM will verify the production details; if any repairs occurred in manufacturing system that will notify to MM. Manufacturing persons will update the work according to the product.

MM

Add M-Persons

See Production Details

See the Repair Details

M- Person will Update the work of Manufacturing

Database

**6. Dealer**

This is the fifth module of this project. In this module dealer will register then login. Then dealer will send the orders to Management team. And receive the products from the management team, and verify those products and update the transportation details.

Dealer

Send Order to MT

Get Products from MT

Verify and Update

Database

**2.2.3 GIVEN INPUT EXPECTED OUTPUT:**

* **User Interface Design**

Input : Enter Login name and Password

Output : If valid user name and password then directly open the home page otherwise show error message and redirect to the registration page.

* **Management Team**

Input : MT Login name and Password

Output: If valid user name and password then directly open the MT home page otherwise show error message and redirect to the MT login page.

* **OEM**

Input : Enter the user name and password

Output : If valid user name and password then directly open the OEM home page otherwise show error message and redirect to the OEM login page.

* **Design Engineer**

Input : Enter the user name and password

Output : If valid user name and password then directly open the DE home page otherwise show error message and redirect to the DE login page.

* **Manufacturing Manager**

Input : Enter the user name and password

Output : If valid user name and password then directly open the MM home page otherwise show error message and redirect to the MM login page.

* **Dealer**

Input : Enter the user name and password

Output: If valid user name and password then directly open the dealer home page otherwise show error message and redirect to the dealer login page.

* 1. **TECHNIQUE USED OR ALGORITHM USED**
* **Iterative Algorithm (IA):**

Integration and coordination of supply-chain functions should be improved. Facilitates the transmission of information between all stakeholders in the supply chain regarding manufacturing, assembling, distribution, and product maintenance. By allowing consumers to and out where a shipment or order is in a given time. With BC, the supply chain may be more trustworthy and transparent.

* **MULAN Model:**

We show that for any network connectivity measures in the SUBLINE family, the connectivity optimization problem with the proposed MULAN model enjoys the diminishing returns property, which naturally lends itself to a family of provable near-optimal optimization algorithms with linear complexity. We unify a family of prevalent network connectivity measures (SUBLINE), which are in close relation to a variety of important network parameters.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

This research aims to see how BC technology may be used to improve current supply chain networks. In this criterion, search using BC Technology, Supply Chain Management, BC Use in SCM, Provenance, and Supply Chain Traceability. Finally, we examined the abstracts and conclusions of the collected publications and determined the relevant techniques and ideas for the proposed survey.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basiras 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 shoulds what the system and not how it should be implemented.

* PROCESSOR : PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
* RAM : 512 MB DD RAM
* MONITOR : 15” COLOR
* HARD DISK : 40 GB

**3.3 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.

* + Front End : J2EE (JSP, SERVLET)
  + Back End : MY SQL 5.5
  + Operating System : Windows 7
  + IDE : Eclipse

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs. The outsourced computation is data is more secured.

* **User**
* Register
* Login
* Send data
* Store the data
* Logout
* **Manager**
* Login
* Add details
* Respond
* View Files details.
* Logout.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

The major non-functional Requirements of the system are as follows.

* **Usability**

The system is designed with completely automated process hence there is no or less user intervention.

* **Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

* **Performance**

This system is developing in the high-level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

* **Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

* **Implementation**

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Struts provides HTML Tag.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**4.1.1 Use Case Diagram**



**EXPLANATION:**

Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use case focus on the behaviour of the system from an external point of view. The identification of actors and use cases results in the definition of the boundary of the system, which is, in differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are outside the boundary of the system, where as the use cases are inside the boundary of the system.

**4.1.2 Class Diagram**



**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification.

**4.1.3 Object Diagram**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.1.4 State Chart Diagram**



**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.1.5 Sequence Diagram**



**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.1.6 Collaboration Diagram**



**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.1.7 Activity Diagram**



**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**4.1.8 Component Diagram**



**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.1.9 E-R Diagram:**

Data base

Verify

Details

Production Details

Update

Add Team Person

Manager

Add DE, MM, and MRO

View Orders

Production Details

Management

Add OEM Person

View Status

Specifications

OEM

View Products

Update Status

Send Orders

Dealer

Verify

Send Specifications

Send to MM

Engineer

**EXPLANATION:**

Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

**4.1.10 Data Flow Diagram:**

**Level 0:**

User Login

User Registration

Login verification

User Page

Error Page

User

Cloud Database

**Level 1:**

MT

Add DE, MM, and MRO

View Production Details

Get Orders and Deliver them

Repair Details

Database

**EXPLANATION:**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

**4.1.11 Deployment Diagram:**



**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger deployment and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination. Results are to be showed to user by data aggregators. All boxes are arrow indicates dependencies.

**4.2 System Architecture**

Add DE, MM

Verify Product & Sales Details

Get Orders from States

Transport to States

Maintain Order Details

Add OEM Person

Send Requirements to OEM

Get Required Material from OEM

Verify with Specifications

Data Base

Get the Issue from MM

Solve the Issue

Send Orders to Management

Get the Ordered Vehicles

Update the Status of Manufacturing

Provide Any Issue in Manufacturing

Get the Status of Manufacturing

Add Maintenance Person

Requirements from DE

Send Requirements to DE

**System Architecture Model**

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 GENERAL**

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA, J2EE and J2ME. In this project J2EE is chosen for implementation.

**5.2 FEATURES OF JAVA**

**5.2.1 THE JAVA FRAMEWORK**

**Java** is a programming language originally developed by James Gosling at Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications the java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

**5.2.2 OBJECTIVES OF JAVA**

To see places of Java in Action in our daily life, explore java.com.

## Why Software Developers Choose Java

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

## Some Ways Software Developers Learn Java

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

**ObjectOriented** To be an Object Oriented language, any language must follow at least the four characteristics.

**1. Inheritance:** It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a feature as needed.

**2. Encapsulation:** It is the mechanism of combining the information and providing the abstraction.

**3. Polymorphism:** As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

**4. Dynamic binding:** Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

**5.2.3 JAVA SWING OVERVIEW**

**Abstract Window Toolkit (AWT) is cross-platform**

Swing provides many controls and widgets to build user interfaces with. Swing class names typically begin with a J such as JButton, JList, JFrame. This is mainly to differentiate them from their AWT counterparts and in general is one-to-one replacements. Swing is built on the concept of Lightweight components vs AWT and SWT's concept of Heavyweight components. The difference between the two is that the Lightweight components are rendered (drawn) using purely Java code, such as drawLine and drawImage, whereas Heavyweight components use the native operating system to render the components.

Some components in Swing are actually heavyweight components. The top-level classes and any derived from them are heavyweight as they extend the AWT versions. This is needed because at the root of the UI, the parent windows need to be provided by the OS. These top-level classes include JWindow, JFrame, JDialog and JApplet. All Swing components to be rendered to the screen must be able to trace their way to a root window of one of those classes.

**Note**: It generally it is not a good idea to mix heavyweight components with lightweight components (other than as previously mentioned) as you will encounter layering issues, e.g., a lightweight component that should appear "on top" ends up being obscured by a heavyweight component. The few exceptions to this include using heavyweight components as the root pane and for popup windows. Generally speaking, heavyweight components will render on top of lightweight components and will not be consistent with the look and feel being used in Swing. There are exceptions, but that is an advanced topic. The truly adventurous may want to consider reading this article from Sun on mixing heavyweight and lightweight components.

**5.2.4 EVOLUTION OF COLLECTION FRAMEWORK:**

Almost all collections in Java are derived from the [**java.util.Collection**](http://download.oracle.com/javase/7/docs/api/java/util/Collection.html) interface. Collection defines the basic parts of all collections. The interface states the add() and remove() methods for adding to and removing from a collection respectively. Also required is the toArray() method, which converts the collection into a simple array of all the elements in the collection. Finally, the contains() method checks if a specified element is in the collection. The Collection interface is a sub interface of [**java.util.Iterable**](http://download.oracle.com/javase/7/docs/api/java/util/Iterable.html), so the iterator() method is also provided. All collections have an iterator that goes through all of the elements in the collection. Additionally, Collection is a generic. Any collection can be written to store any class. For example, Collection<String> can hold strings, and the elements from the collection can be used as strings without any casting required.

There are three main types of collections:

* Lists: always ordered, may contain duplicates and can be handled the same way as usual arrays
* Sets: cannot contain duplicates and provide random access to their elements
* Maps: connect unique keys with values, provide random access to its keys and may host duplicate values

**LIST:**

Lists are implemented in the JCF via the java.util.List interface. It defines a list as essentially a more flexible version of an array. Elements have a specific order, and duplicate elements are allowed. Elements can be placed in a specific position. They can also be searched for within the list. Two concrete classes implement List. The first is java.util.ArrayList, which implements the list as an array. Whenever functions specific to a list are required, the class moves the elements around within the array in order to do it. The other implementation is java.util.LinkedList. This class stores the elements in nodes that each have a pointer to the previous and next nodes in the list. The list can be traversed by following the pointers, and elements can be added or removed simply by changing the pointers around to place the node in its proper place.

**SET:**

Java's [java.util.Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html) interface defines the set. A set can't have any duplicate elements in it. Additionally, the set has no set order. As such, elements can't be found by index. Set is implemented by java.util.HashSet, java.util.LinkedHashSet, and java.util.TreeSet. HashSet uses a hash table. More specifically, it uses a [java.util.HashMap](http://download.oracle.com/javase/7/docs/api/java/util/HashMap.html) to store the hashes and elements and to prevent duplicates. Java.util.LinkedHashSet extends this by creating a doubly linked list that links all of the elements by their insertion order. This ensures that the iteration order over the set is predictable. [java.util.TreeSet](http://download.oracle.com/javase/7/docs/api/java/util/TreeSet.html) uses a red-black tree implemented by a [java.util.TreeMap](http://download.oracle.com/javase/7/docs/api/java/util/TreeMap.html). The red-black tree makes sure that there are no duplicates. Additionally, it allows Tree Set to implement java.util.SortedSet.

The [java.util.Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html) interface is extended by the java.util.SortedSet interface. Unlike a regular set, the elements in a sorted set are sorted, either by the element's compareTo() method, or a method provided to the constructor of the sorted set. The first and last elements of the sorted set can be retrieved, and subsets can be created via minimum and maximum values, as well as beginning or ending at the beginning or ending of the sorted set. The SortedSet interface is implemented by java.util.TreeSet

[java.util.SortedSet](http://download.oracle.com/javase/7/docs/api/java/util/SortedSet.html) is extended further via the java.util.NavigableSet interface. It's similar to SortedSet, but there are a few additional methods. The floor(), ceiling(), lower(), and higher() methods find an element in the set that's close to the parameter. Additionally, a descending iterator over the items in the set is provided. As with SortedSet, java.util.TreeSet implements NavigableSet.

**MAP:**

Maps are defined by the java.util.Map interface in Java. Maps are simple data structures that associate a key with a value. The element is the value. This lets the map be very flexible. If the key is the hash code of the element, the map is essentially a set. If it's just an increasing number, it becomes a list. Maps are implemented by java.util.HashMap, java.util.LinkedHashMap, and java.util.TreeMap. HashMap uses a hash table. The hashes of the keys are used to find the values in various buckets. LinkedHashMap extends this by creating a doubly linked list between the elements. This allows the elements to be accessed in the order in which they were inserted into the map. TreeMap, in contrast to HashMap and LinkedHashMap, uses a red-black tree. The keys are used as the values for the nodes in the tree, and the nodes point to the values in the map

**THREAD:**

Simply put, a threadis a program's path of execution. Most programs written today run as a single thread, causing problems when multiple events or actions need to occur at the same time. Let's say, for example, a program is not capable of drawing pictures while reading keystrokes. The program must give its full attention to the keyboard input lacking the ability to handle more than one event at a time. The ideal solution to this problem is the seamless execution of two or more sections of a program at the same time.

**CREATING THREADS:**

Java's creators have graciously designed two ways of creating threads: implementing an interface and extending a class. Extending a class is the way Java inherits methods and variables from a parent class. In this case, one can only extend or inherit from a single parent class. This limitation within Java can be overcome by implementing interfaces, which is the most common way to create threads. (Note that the act of inheriting merely allows the class to be run as a thread. It is up to the class to start() execution, etc.)

Interfaces provide a way for programmers to lay the groundwork of a class. They are used to design the requirements for a set of classes to implement. The interface sets everything up, and the class or classes that implement the interface do all the work. The different set of classes that implement the interface have to follow the same rules.

**5.3 CONCLUSION**

Swing's high level of flexibility is reflected in its inherent ability to override the native host [operating system](http://en.wikipedia.org/wiki/Operating_system) (OS)'s GUI controls for displaying itself. Swing "paints" its controls using the Java 2D APIs, rather than calling a native user interface toolkit. The Java thread scheduler is very simple. All threads have a priority value which can be changed dynamically by calls to the threads setPriority() method . Implementing the above concepts in our project to do the efficient work among the Server.

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

The Implementation is nothing but sores code of project.

**6.2 IMPLEMENTATION**

**Coding:**

**Marble.java**

package com.control;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.dao.Dao;

@WebServlet("/Marble")

public class Marble extends HttpServlet {

private static final long serialVersionUID = 1L;

public Marble() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter o = response.getWriter();

String uid = request.getParameter("uid");

String pwd = request.getParameter("pwd");

String sql = "select \* from marble where email='"+uid+"' and password='"+pwd+"'";

boolean b = Dao.getData(sql);

HttpSession session = request.getSession();

if(b == true){

sql = "select marble from marble where email='"+uid+"'";

String name = Dao.getName(sql);

session.setAttribute("uid", uid);

session.setAttribute("name", name);

response.sendRedirect("mhome.jsp");

}else{

o.println("<script type=\"text/javascript\">");

o.println("alert('Please enter valid Details');");

o.println("window.location='marble.jsp';</script>");

}

}

}

**UserReg.java**

package com.control;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.bean.UserBean;

import com.dao.Dao;

@WebServlet("/UserReg")

public class UserReg extends HttpServlet {

private static final long serialVersionUID = 1L;

public UserReg() {

super();

} protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

PrintWriter o = response.getWriter();

String name = request.getParameter("name");

String email = request.getParameter("uid");

String pwd = request.getParameter("pwd");

String mob = request.getParameter("mob");

String marble = request.getParameter("marble");

String loc = request.getParameter("loc");

String sql = "insert into user values(?,?,?,?,?,?,?)";

UserBean ub = new UserBean();

ub.setName(name);

ub.setEmail(email);

ub.setPwd(pwd);

ub.setMob(mob);

ub.setMarble(marble);

ub.setLoc(loc);

int i = Dao.setUser(sql, ub);

if(i > 0){

o.println("<script type=\"text/javascript\">");

o.println("alert('Register Successfully...');");

o.println("window.location='user.jsp';</script>");

}else{

o.println("<script type=\"text/javascript\">");

o.println("alert('Please enter valid Details/Already Exist');");

o.println("window.location='userreg.jsp';</script>");

}

}

}

**User.java**

package com.control;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.dao.Dao;

@WebServlet("/User")

public class User extends HttpServlet {

private static final long serialVersionUID = 1L;

public User() {

super();

} protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

} protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter o = response.getWriter();

String uid = request.getParameter("uid");

String pwd = request.getParameter("pwd");

String sql = "select \* from user where email='"+uid+"' and password='"+pwd+"'";

boolean b = Dao.getData(sql);

HttpSession session = request.getSession();

if(b == true){

session.setAttribute("uid", uid);

response.sendRedirect("uhome.jsp");

}else{

o.println("<script type=\"text/javascript\">");

o.println("alert('Please enter valid Details');");

o.println("window.location='user.jsp';</script>");

}

}

}

**BC.java**

package com.control;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/BC")

public class BC extends HttpServlet {

private static final long serialVersionUID = 1L;

public BC() {

super();

} protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

} protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter o = response.getWriter();

String uid = request.getParameter("uid");

String pwd = request.getParameter("pwd");

if(uid.equalsIgnoreCase("bc") && pwd.equals("bc")){

response.sendRedirect("bchome.jsp");

}else{

o.println("<script type=\"text/javascript\">");

o.println("alert('Please enter valid Details');");

o.println("window.location='bc.jsp';</script>");

}

}

}

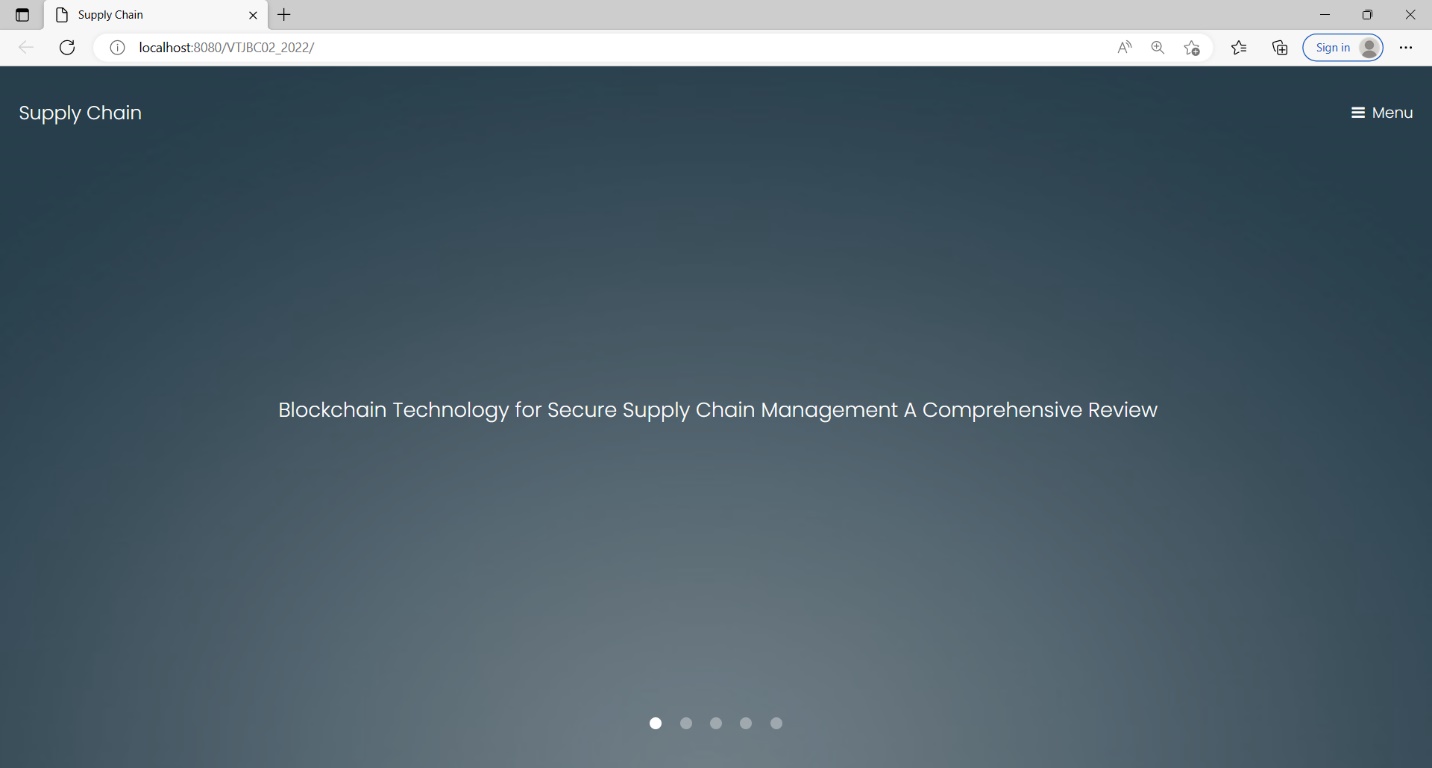
**CHAPTER 7**

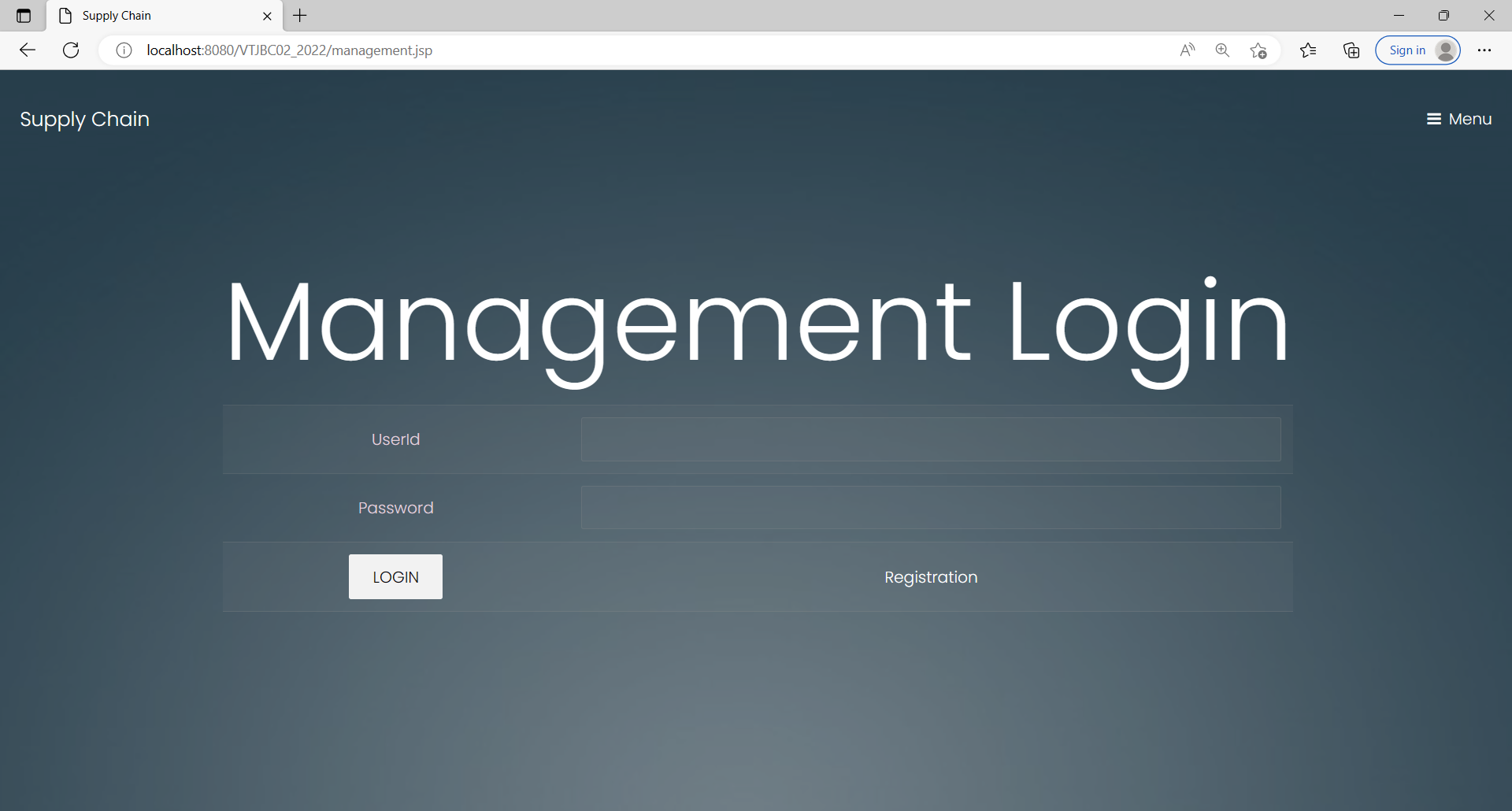
**SNAPSHOTS**

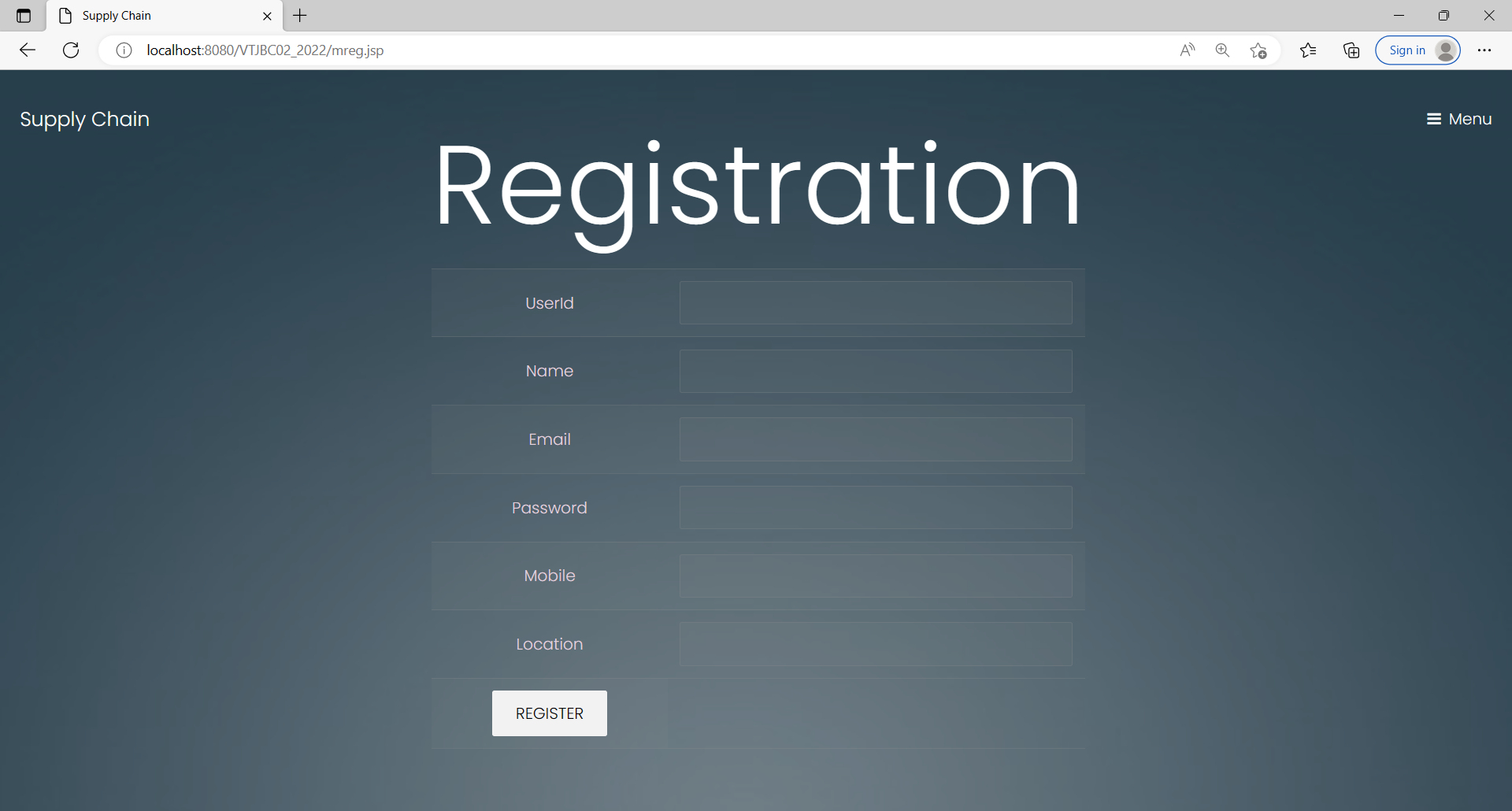
**7.1 GENERAL**

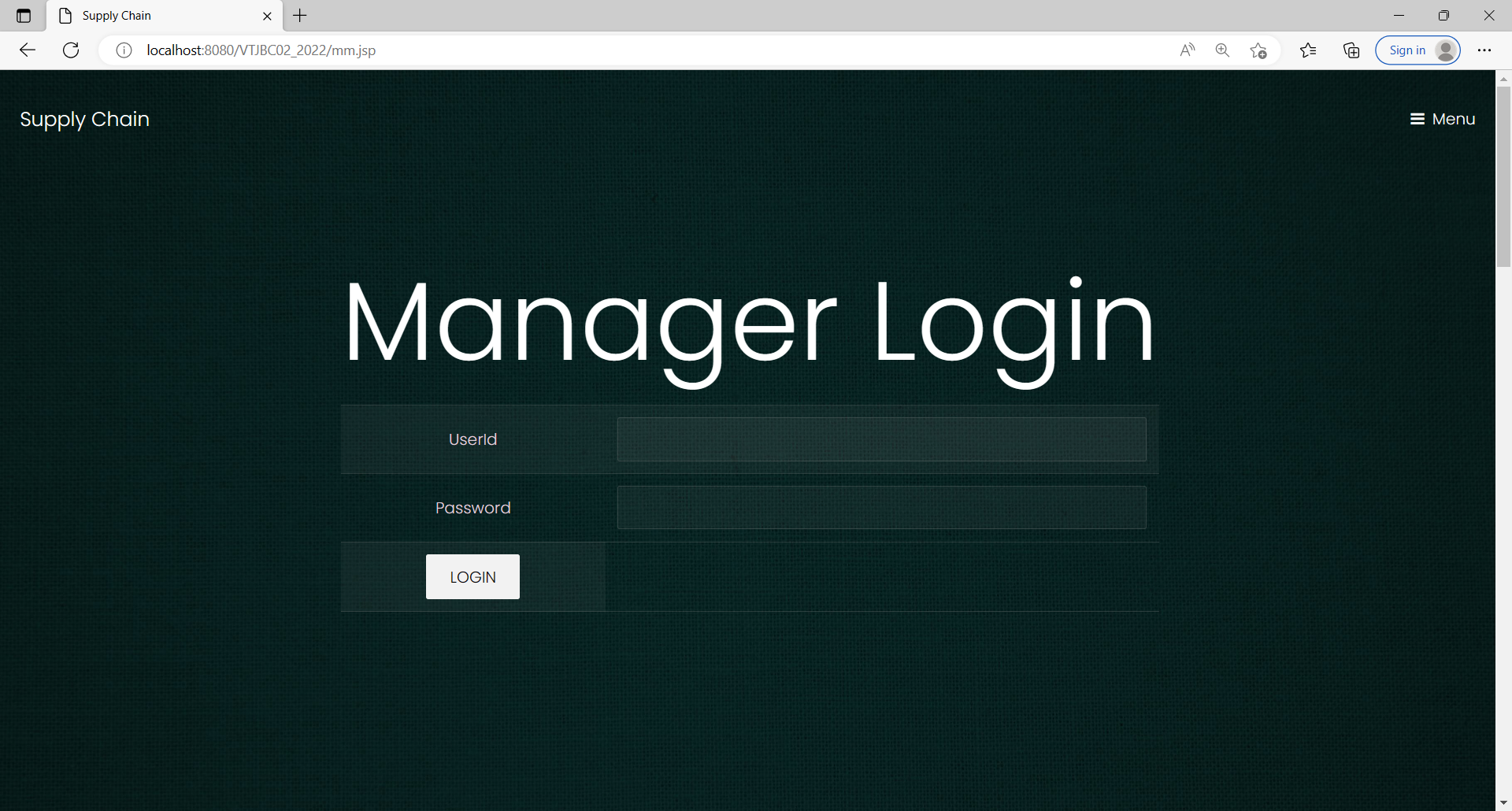
This project is implements like web application using COREJAVA and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

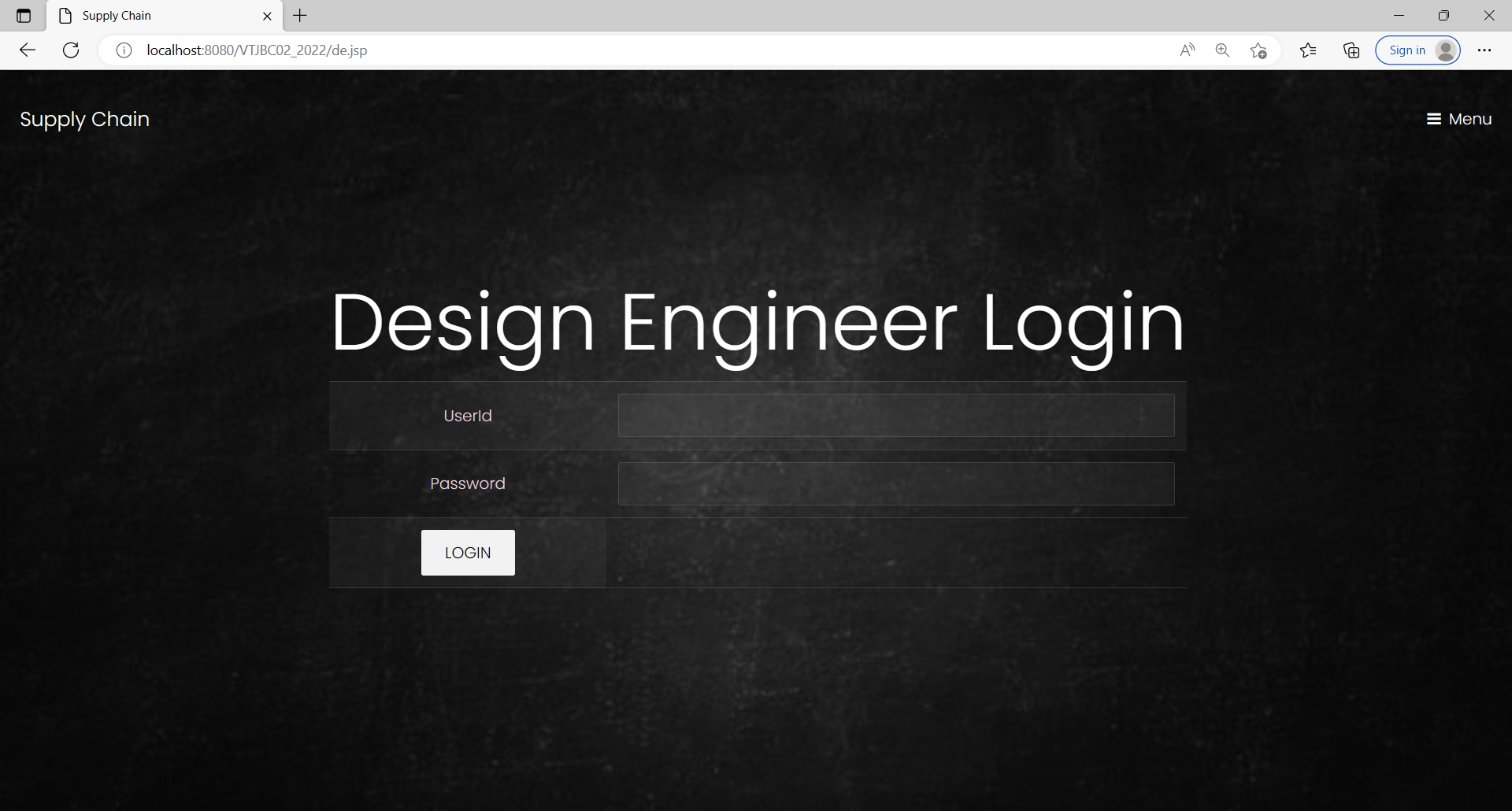
**7.2 VARIOUS SNAPSHOTS**











**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3 Types of Tests**

**8.3.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 Functional Test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance Testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.3.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**TEST CASES:**

Test cases can be divided in to two types. First one is Positive test cases and second one is negative test cases. In positive test cases are conducted by the developer intention is to get the output. In negative test cases are conducted by the developer intention is to don’t get the output.

**TEST PLAN**

The test procedure is started by building up a thorough arrangement to test the general usefulness and extraordinary highlights on an assortment of stage mixes. Exacting quality control methods are utilized. The procedure checks that the application meets the necessities indicated in the framework prerequisites report and is sans bug.

Any project can be separated into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing serves to character the potential bugs in the individual segment, so the segment that has bugs can be recognized and can be redressed from mistakes.

**CHAPTER 9**

**APPLICATION**

**9.1 GENERAL**

The developers need to be extremely cautious in designing the complex smart contacts, taking all those codes into account, which makes the smart contact cumbersome. The present storage costs connected with BC are quite high, with 1MB of data costing 550 USD. So, to adopt BC-based solutions, the cost of storage must be minimized. Cooperation, cooperative collaboration, and proper consensus methods would be critical components in implementing BC for SCM in practice. More strong consensus algorithms and security enforcing methods are required. Reward all supply chain players with incentives to correctly follow the BC-based supply chain protocols. The incentive-based technique of rewarding will pave the path for a long-term, stable future [66]. Altering the exact implementation of BC protocols may solve the scalability problem, such as Lightning Network and State Channels, which maintain the core BC protocols but perform transactions off-chain. We may also utilize BC sharding, which allows each node to handle only a portion of transactions, decreasing node calculation and storage and improving performance. Another possibility is to use a digital currency on the BC network to fund the supply chain.

**9.2 FUTURE ENHANCEMENT**

Furthermore, the obstacles, possibilities, and future perspectives of using BC in the supply chain are thoroughly examined to identify open research questions on utilizing BC realistically in the future. There is a lack of in-depth understanding of BC technology, which might jeopardize its benefits. We think that our study gives academics, engineers, educators, and general readers a proper orientation on the theoretical insights of BC. It also identifies future study goals in areas that combine future technology and BC.

**CHAPTER 10**

**CONCLUSION & REFERENCE**

**10.1 CONCLUSION**

This study attempts to investigate briefly how BC technology works and when it should be utilized to address supply chain challenges. BC technology is employed in SCM in a variety of sectors. The present state of use of BC and Smart Contracts in numerous major industrial domains is studied in this paper. The survey delivers academically sound data on the overall state of BC deployment for various supply chains. The study’s findings and conclusions show that research on BC-based supply chains is a growing topic garnering a lot of attention. The majority of the reviewed papers that were evaluated agreed on the prospective benefits that BC may offer to the supply chain.

**10.2 REFERENCE**

[1] M. J. Labaran and M. Hamma- Adama, ‘‘The Nigerian pharmaceutical supply chain: Blockchain adoption, counterfeit drugs and successful deployment of COVID-19 vaccine in Nigeria,’’ J. Sci. Res. Rep., vol. 27, no. 2, pp. 20–36, Mar. 2021, doi: 10.9734/jsrr/2021/v27i230356.

[2] M. Sharma and G. Sikka, ‘‘Blockchain based approaches for preventing drug counterfeit: A survey,’’ Int. J. Eng. Res. Technol., vol. 9, no. 8, pp. 1–6, 2021.

[3] A. Batwa and A. Norrman, ‘‘Blockchain technology and trust in supply chain management: A literature review and research agenda,’’ Oper. Supply Chain Manage., Int. J., vol. 14, no. 2, pp. 203–220, Jan. 2021.

[4] R. Raman, J. Sushmitha, and M. K. Nalini, ‘‘A survey paper on blockchain technologies in supply chain management,’’ Int. J. Res. Eng. Sci., vol. 9, no. 6, pp. 79–86, 2021.

[5] Z. Li, R. Y. Zhong, Z.-G. Tian, H.-N. Dai, A. V. Barenji, and G. Q. Huang, ‘‘Industrial blockchain: A state-of-the-art survey,’’ Robot. Comput. Integr. Manuf., vol. 70, Aug. 2021, Art. no. 102124.

[6] S. E. Chang and Y. Chen, ‘‘When blockchain meets supply chain: A systematic literature review on current development and potential applications,’’ IEEE Access, vol. 8, pp. 62478–62494, 2020.

[7] S. Tönnissen and F. Teuteberg, ‘‘Analysing the impact of blockchain-technology for operations and supply chain management: An explanatory model drawn from multiple case studies,’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 101953.

[8] N. Vadgama and P. Tasca, ‘‘An analysis of blockchain adoption in supply chains between 2010 and 2020,’’ Frontiers Blockchain, vol. 4, Mar. 2021, Art. no. 610476.

[9] P. Gonczol, P. Katsikouli, L. Herskind, and N. Dragoni, ‘‘Blockchain implementations and use cases for supply chains—A survey,’’ IEEE Access, vol. 8, pp. 11856–11871, 2020.n. 2013.

[10] S. Shingh, V. Kamalvanshi, S. Ghimire, and S. Basyal, ‘‘Dairy supply chain system based on blockchain technology,’’ Asian J. Econ., Bus. Accounting, vol. 14, pp. 13–19, Mar. 2020.

[11] S. Kummer, D. M. Herold, M. Dobrovnik, J. Mikl, and N. Schäfer, ‘‘A systematic review of blockchain literature in logistics and supply chain management: Identifying research questions and future directions,’’ Future Internet, vol. 12, no. 3, p. 60, Mar. 2020.

[12] Y. Zou, T. Meng, P. Zhang, W. Zhang, and H. Li, ‘‘Focus on blockchain: A comprehensive survey on academic and application,’’ IEEE Access, vol. 8, pp. 187182–187201, 2020.

[13] N. Hackius and M. Petersen, ‘‘Translating high hopes into tangible bene-ﬁts: How incumbents in supply chain and logistics approach blockchain,’’ IEEE Access, vol. 8, pp. 34993–35003, 2020.

[14] M. Mamtashanti, J. Rahul, and T. Kashyap, ‘‘A review on regulatory requirements to prevent counterfeit drugs in India,’’ Int. J. Pharmaceutical Invest., vol. 10, no. 3, pp. 257–262, 2020. [Online]. Available: https://www.jpionline.org/index.php/ijpi/article/view/704/463

[15] V. Ahmadi, S. Benjelloun, M. E. Kik, T. Sharma, H. Chi, and W. Zhou, ‘‘Drug governance: IoT-based blockchain implementation in the pharmaceutical supply chain,’’ in Proc. 6th Int. Conf. Mobile Secure Services (MobiSecServ), Feb. 2020, pp. 1–8.

[16] A. Liu, T. Liu, J. Mou, and R. Wang, ‘‘A supplier evaluation model based on customer demand in blockchain tracing anti-counterfeiting platform project management,’’ J. Manage. Sci. Eng., vol. 5, no. 3, pp. 172–194, Sep. 2020.

[17] S. Dede, M. C. Köseoğlu, and H. F. Yercan, ‘‘Learning from early adopters of blockchain technology: A systematic review of supply chain case studies,’’ Technol. Innov. Manage. Rev., vol. 2021, pp. 19–31, Jul. 2021.

[18] Q. Song, Y. Chen, Y. Zhong, K. Lan, S. Fong, and R. Tang, ‘‘A supply-chain system framework based on Internet of Things using blockchain technology,’’ ACM Trans. Internet Technol., vol. 21, no. 1, pp. 1–24, Feb. 2021.

[19] H. Hellani, L. Sliman, A. E. Samhat, and E. Exposito, ‘‘On blockchain integration with supply chain: Overview on data transparency,’’ Logistics, vol. 5, no. 3, p. 46, Jul. 2021.

[20] S. Yadav and S. P. Singh, ‘‘Blockchain critical success factors for sustain-able supply chain,’’ Resour., Conservation Recycling, vol. 152, Jan. 2020, Art. no. 104505.

[21] J. Moosavi, ‘‘Blockchain in supply chain management: A review, bibliometric, and network analysis,’’ Environ. Sci. Pollut. Res., vol. 2021, pp. 1–15, Feb. 2021.

[22] H. Min, ‘‘Blockchain technology for enhancing supply chain resilience,’’ Bus. Horizons, vol. 62, no. 1, pp. 35–45, Jan. 2019.

[23] R. Cole, M. Stevenson, and J. Aitken, ‘‘Blockchain technology: Implications for operations and supply chain management,’’ Supply Chain Manage., Int. J., vol. 24, no. 4, pp. 469–483, Jun. 2019.

[24] R. Azzi, R. K. Chamoun, and M. Sokhn, ‘‘The power of a blockchain-based supply chain,’’ Comput. Ind. Eng., vol. 135, pp. 582–592, Sep. 2019.

[25] B. Yong, J. Shen, X. Liu, F. Li, H. Chen, and Q. Zhou, ‘‘An intelligent blockchain-based system for safe vaccine supply and supervision,’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 102024.

[26] S. Saberi, M. Kouhizadeh, and J. Sarkis, ‘‘Blockchains and the supply chain: Findings from a broad study of practitioners,’’ IEEE Eng. Manag. Rev., vol. 47, no. 3, pp. 95–103, Sep. 2019.

[27] G. Blossey, J. Eisenhardt, and G. Hahn, ‘‘Blockchains technology in supply chain management: An application perspective,’’ in Proc. 52nd Hawaii Int. Conf. Syst. Sci., 2019.

[28] J. Kolb, ‘‘Industrial application of blockchains technology—Erasing the weaknesses of vendor managed inventory,’’ in Proc. 26th Eur. Conf. Inf. Syst., 2018, pp. 1–17.

[29] N. Kshetri, ‘‘Blockchain and sustainable supply chain management in developing countries,’’ Int. J. Inf. Manage. vol. 60, Oct. 2021, Art. no. 102376.

[30] N. Kshetri, ‘‘1 blockchain’s roles in meeting key supply chain management objectives,’’ Int. J. Inf. Manage., vol. 39, pp. 80–89, Jan. 2018.

[31] Z. Liu and Z. Li, ‘‘A blockchain-based framework of cross-border e- commerce supply chain,’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 102059.

[32] K. Behnke and M. F. W. H. A. Janssen, ‘‘Boundary conditions for trace-ability in food supply chains using blockchain technology,’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 101969.

[33] C. Antal, T. Cioara, M. Antal, and I. Anghel, ‘‘Blockchain platform for COVID-19 vaccine supply management,’’ IEEE Open J. Comput. Soc., vol. 2, pp. 164–178, 2021, doi: 10.1109/OJCS.2021.3067450.

[34] Q. Lin, H. Wang, X. Pei, and J. Wang, ‘‘Food safety traceability system based on blockchain and EPCIS,’’ IEEE Access, vol. 7, pp. 20698–20707, 2019, doi: 10.1109/ACCESS.2019.2897792.

[35] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, ‘‘A blockchain-based approach for drug traceability in healthcare supply chain,’’ IEEE Access, vol. 9, pp. 9728–9743, 2021, doi: 10.1109/ACCESS.2021.3049920.

[36] F. Dietrich, ‘‘Review and analysis of blockchain projects in supply chain management,’’ Proc. Comput. Sci., vol. 180, Jan. 2021, pp. 724–733.

[37] X. Yang, M. Li, H. Yu, M. Wang, D. Xu, and C. Sun, ‘‘A trusted blockchain-based traceability system for fruit and vegetable agricultural products,’’ IEEE Access, vol. 9, pp. 36282–36293, 2021, doi: 10.1109/ACCESS.2021.3062845.

[38] L.-W. Wong, L.-Y. Leong, J.-J. Hew, G. W.-H. Tan, and K.-B. Ooi, ‘‘Time to seize the digital evolution: Adoption of blockchain in operations and supply chain management among Malaysian SMEs,’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 101997.

[39] S. E. Chang, Y.-C. Chen, and M.-F. Lu, ‘‘Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process,’’ Technol. Forecasting Social Change, vol. 144, pp. 1–11, Jul. 2019.

[40] F. Tian, ‘‘A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things,’’ in Proc. Int. Conf. Service Syst. Service Manage., Jun. 2017, pp. 1–6.

[41] J. Hao and Y. S. H. Luo, ‘‘A safe and efﬁcient storage scheme based on blockchain and IPFS for agricultural products tracking,’’ J. Comput., vol. 29, no. 6, pp. 158–167, 2018.

[42] G. Baralla, S. Ibba, M. Marchesi, R. Tonelli, and S. Missineo, ‘‘A blockchain based system to ensure transparency and reliability in food supply chain,’’ in Proc. Eur. Conf. Parallel Process., Dec. 2018, pp. 379–391, doi: 10.1007/978-3-030-10549-5\_30.

[43] G. Baralla, A. Pinna, and G. Corrias, ‘‘Ensure traceability in European food supply chain by using a blockchain system,’’ in Proc. IEEE/ACM 2nd Int. Workshop Emerg. Trends Softw. Eng. Blockchain (WETSEB), May 2019, pp. 40–47.

[44] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, ‘‘Blockchain-based soybean traceability in agricultural supply chain,’’ IEEE Access, vol. 7, pp. 73295–73305, 2019, doi: 10.1109/ACCESS.2019.2918000.

[45] A. Shahid, A. Almogren, N. Javaid, F. A. Al-Zahrani, M. Zuair, and M. Alam, ‘‘Blockchain-based agri-food supply chain: A complete solution,’’ IEEE Access, vol. 8, pp. 69230–69243, 2020.

[46] A. Shahid, ‘‘Blockchain-based reputation system in agri-food supply chain,’’ in Proc. Int. Conf. Adv. Inf. Netw. Appl., 2020, pp. 12–21.

[47] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, and Y. Chen, ‘‘Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry,’’ Comput. Ind. Eng., vol. 154, Apr. 2021, Art. no. 107130.

[48] M.-J. Lee, J.-T. Luo, J.-J. Shao, and N.-F. Huang, ‘‘A trustworthy food resume traceability system based on blockchain technology,’’ in Proc. Int. Conf. Inf. Netw. (ICOIN), Jan. 2021, pp. 546–552.

[49] R. Zheng, J. Jiang, X. Hao, W. Ren, F. Xiong, and Y. Ren, ‘‘BcBIM: A blockchain-based big data model for BIM modiﬁcation audit and provenance in mobile cloud,’’ Math. Problems Eng., vol. 2019, pp. 1–13, Mar. 2019.

[50] G. Perboli, S. Musso, and M. Rosano, ‘‘Blockchain in logistics and supply chain: A lean approach for designing real-world use cases,’’ IEEE Access, vol. 6, pp. 62018–62028, 2018.

[51] I. A. Omar, M. Debe, R. Jayaraman, K. Salah, M. Omar, and J. Arshad, ‘‘Blockchain-based supply chain traceability for COVID-19 personal protective equipment,’’ Comput. Ind. Eng., vol. 167, May 2022, Art. no. 107995, doi: 10.1016/j.cie.2022.107995.

[52] R. Singh, A. D. Dwivedi, and G. Srivastava, ‘‘Internet of Things based blockchain for temperature monitoring and counterfeit pharmaceutical prevention,’’ Sensors, vol. 20, no. 14, p. 3951, Jul. 2020.

[53] N. Alzahrani and N. Bulusu, ‘‘Block-supply chain: A new anti-counterfeiting supply chain using NFC and blockchain,’’ in Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst., Jun. 2018, pp. 30–35.

[54] K. Abbas, M. Afaq, T. Ahmed Khan, and W.-C. Song, ‘‘A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharmaceutical industry,’’ Electronics, vol. 9, no. 5, p. 852, May 2020, doi: 10.3390/ELECTRONICS9050852.

[55] R. Kumar and R. Tripathi, ‘‘Traceability of counterfeit medicine supply chain through blockchain,’’ in Proc. 11th Int. Conf. Commun. Syst. Netw. (COMSNETS), Jan. 2019, pp. 568–570.

[56] M. C. Jayaprasanna, V. A. Soundharya, M. Suhana, and S. Sujatha, ‘‘A block chain based management system for detecting counterfeit prod-uct in supply chain,’’ in Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV), Feb. 2021, pp. 253–257.

[57] M. Uddin, K. Salah, R. Jayaraman, S. Pesic, and S. Ellahham, ‘‘Blockchain for drug traceability: Architectures and open challenges,’’ Health Informat. J., vol. 27, no. 2, Apr. 2021, Art. no. 146045822110112.

[58] V. Sudha and R. P. K. Shanmughasundaram, ‘‘Blockchain based solution to improve the supply chain management in Indian agriculture,’’ in Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS), Mar. 2021, pp. 1289–1292.

[59] R. E. Spekman, J. W. Kamauff, and N. Myhr, ‘‘An empirical investigation into supply chain management: A perspective on partnerships,’’ Supply Chain Manage., Int. J., vol. 3, no. 2, pp. 53–67, Jun. 1998.

[60] R. C. Mayer, J. H. Davis, and F. D. Schoorman, ‘‘An integrative model of organizational trust,’’ Acad. Manage. Rev., vol. 20, no. 3, pp. 709–734, 1995.

[61] J. Hua, X. Wang, M. Kang, H. Wang, and F.-Y. Wang, ‘‘Blockchain based provenance for agricultural products: A distributed platform with duplicated and shared bookkeeping,’’ in Proc. IEEE Intell. Vehicles Symp.(IV), Jun. 2018, pp. 97–101.

[62] H. Kumiawan, J. Kim, and H. Ju, ‘‘Utilization of the blockchain network in the public community health center medicine supply chain,’’ in Proc. 21st Asia–Paciﬁc Netw. Oper. Manage. Symp. (APNOMS), Sep. 2020, pp. 235–238, doi: 10.23919/APNOMS50412.2020.9237042.

[63] J. Aslam, A. Saleem, N. T. Khan, and Y. B. Kim, ‘‘Factors inﬂuencing blockchain adoption in supply chain management practices: A study based on the oil industry,’’ J. Innov. Knowl., vol. 6, no. 2, pp. 124–134, 2021.

[64] R. Alkhudary, X. Brusset, and P. Fenies, ‘‘Blockchain and risk in sup-ply chain management,’’ in Dynamics in Logistics. Cham, Germany: Springer, 2020, pp. 159–165.

[65] P. Dutta, T. M. Choi, S. Somani, and R. Butala, ‘‘Blockchain technology in supply chain operations: Applications, challenges and research opportunities,’’ Transp. Res. E, Logistics Transp. Rev., vol. 142, Oct. 2020, Art. no. 102067.

[66] S. A. Pranesh, V. Kannan, V. N. Viswanathan, and M. Vijayalakshmi, ‘‘Design and analysis of incentive mechanism for ethereum-based supply chain management systems,’’ in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–6.

[67] M. H. Ronaghi, ‘‘A blockchain maturity model in agricultural supply chain,’’ Inf. Process. Agricult., vol. 8, pp. 398–408, Sep. 2020.

[68] V. Babich and G. Hilary, ‘‘OM forum—Distributed ledgers and operations: What operations management researchers should know about blockchain technology,’’ Manuf. Service Oper. Manage., vol. 22, no. 2, pp. 223–240, Mar. 2020.

[69] L. Wang, L. Xu, Z. Zheng, S. Liu, X. Li, L. Cao, J. Li, and C. Sun, ‘‘Smart contract-based agricultural food supply chain traceability,’’ IEEE Access, vol. 9, pp. 9296–9307, 2021, doi: 10.1109/ACCESS.2021.3050112.

[70] H. Chen, Z. Chen, F. Lin, and P. Zhuang, ‘‘Effective management for blockchain-based agri-food supply chains using deep reinforcement learning,’’ IEEE Access, vol. 9, pp. 36008–36018, 2021, doi: 10.1109/ACCESS.2021.3062410.

[71] A. Iftekhar and X. Cui, ‘‘Blockchain-based traceability system that ensures food safety measures to protect consumer safety and COVID-19 free supply chains,’’ Foods, vol. 10, no. 6, p. 1289, Jun. 2021.

[72] D. Bumblauskas, A. Mann, B. Dugan, and J. Rittmer, ‘‘A blockchain use case in food distribution: Do you know where your food has been?’’ Int. J. Inf. Manage., vol. 52, Jun. 2020, Art. no. 102008.

[73] K. Gao, Y. Liu, H. Xu, and T. Han, ‘‘Hyper-FTT: A food supply-chain trading and traceability system based on hyperledger fabric,’’ in Proc. Int. Conf. Blockchain Trustworthy Syst., 2020, pp. 648–661, doi: 10.1007/978-981-15-2777-7\_53.

[74] X. Zhang, P. Sun, J. Xu, X. Wang, J. Yu, Z. Zhao, and Y. Dong, ‘‘Blockchain-based safety management system for the grain sup-ply chain,’’ IEEE Access, vol. 8, pp. 36398–36410, 2020, doi: 10.1109/ACCESS.2020.2975415.

[75] T. Ferdousi, D. Gruenbacher, and C. M. Scoglio, ‘‘A permissioned dis-tributed ledger for the US beef cattle supply chain,’’ IEEE Access, vol. 8, pp. 154833–154847, 2020.

[76] I. G. M. T. Pradana, T. Djatna, and I. Hermadi, ‘‘Blockchain modeling for traceability information system in supply chain of coffee agroindustry,’’ in Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS), Oct. 2020, pp. 217–224.

[77] R. Ekawati, Y. Arkeman, S. Suprihatin, and T. C. Sunarti, ‘‘Design of intelligent decision support system for sugar cane supply chains based on blockchain technology,’’ in Proc. 2nd Int. Conf. Ind. Electr. Electron. (ICIEE), Oct. 2020, pp. 153–157.

[78] S. Mondal, K. P. Wijewardena, S. Karuppuswami, N. Kriti, D. Kumar, and P. Chahal, ‘‘Blockchain inspired RFID-based information architecture for food supply chain,’’ IEEE Internet Things J., vol. 6, no. 3, pp. 5803–5813, Jun. 2019, doi: 10.1109/JIOT.2019.2907658.

[79] H. Huang, X. Zhou, and J. Liu, ‘‘Food supply chain traceability scheme based on blockchain and EPC technology,’’ in Smart Blockchain (Lecture Notes in Computer Science), vol. 11911, M. Qiu, Ed. Cham, Switzerland: Springer, 2019, pp. 32–42, doi: 10.1007/978-3-030-34083-4\_4.

[80] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, ‘‘Blockchain-driven IoT for food traceability with an integrated consensus mechanism,’’ IEEE Access, vol. 7, pp. 129000–129017, 2019, doi: 10.1109/ACCESS.2019.2940227.

[81] W. Sun, X. Zhu, T. Zhou, Y. Su, and B. Mo, ‘‘Application of blockchain and RFID in anti-counterfeiting traceability of liquor,’’ in Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC), Dec. 2019, pp. 1248–1251.

[82] A. Arena, A. Bianchini, P. Perazzo, C. Vallati, and G. Dini, ‘‘BRUSCHETTA: An IoT blockchain-based framework for certifying extra virgin olive oil supply chain,’’ in Proc. IEEE Int. Conf. Smart Com-put. (SMARTCOMP), Jun. 2019, pp. 173–179, doi: 10.1109/SMART-COMP.2019.00049.

[83] D. Bechtsis, N. Tsolakis, A. Bizakis, and D. Vlachos, ‘‘A blockchain framework for containerized food supply chains,’’ Comput. Aided Chem. Eng., vol. 46, Jan. 2019, pp. 1369–1374.

[84] A. Jaiswal, S. Chandel, A. Muzumdar, G. M. Madhu, C. Modi, and C. Vyjayanthi, ‘‘A conceptual framework for trustworthy and incentivized trading of food grains using distributed ledger and smart con-tracts,’’ in Proc. IEEE 16th India Council Int. Conf. (INDICON), Dec. 2019, pp. 1–4.

[85] R. C. Koirala, K. Dahal, S. Matalonga, and R. Rijal, ‘‘A supply chain model with blockchain-enabled reverse auction bidding process for transparency and efﬁciency,’’ in Proc. 13th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA), Aug. 2019, pp. 1–6.

[86] S. Madumidha, ‘‘A theoretical implementation: Agriculture-food supply chain management using blockchain technology,’’ in Proc. TEQIP III Sponsored Int. Conf. Microw. Integr. Circuits, Photon. Wireless Netw. (IMICPW), May 2019, pp. 174–178.

[87] H. Moudoud, S. Cherkaoui, and L. Khoukhi, ‘‘An IoT blockchain architecture using oracles and smart contracts: The use-case of a food supply chain,’’ in Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC), Sep. 2019, pp. 1–6.

[88] G. Subramanian, A. S. Thampy, N. V. Ugwuoke, and B. Ramnani, ‘‘Crypto pharmacy—Digital medicine: A mobile application integrated with hybrid blockchain to tackle the issues in pharma supply chain,’’ IEEE Open J. Comput. Soc., vol. 2, pp. 26–37, 2021, doi: 10.1109/OJCS.2021.3049330.

[89] O. Alkhoori, A. Hassan, O. Almansoori, M. Debe, K. Salah, R. Jayaraman, J. Arshad, and M. H. U. Rehman, ‘‘Design and implementation of CryptoCargo: A blockchain-powered smart shipping container for vaccine distribution,’’ IEEE Access, vol. 9, pp. 53786–53803, 2021, doi: 10.1109/ACCESS.2021.3070911.

[90] Z. Qiu and Y. Zhu, ‘‘A novel structure of blockchain applied in vac-cine quality control: Double-chain structured blockchain system for vac-cine anticounterfeiting and traceability,’’ J. Healthcare Eng., vol. 2021, pp. 1–10, Mar. 2021.

[91] R. W. Ahmad, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar, and S. Ellahham, ‘‘Blockchain-based forward supply chain and waste management for COVID-19 medical equipment and supplies,’’ IEEE Access, vol. 9, pp. 44905–44927, 2021, doi: 10.1109/ACCESS.2021.3066503.

[92] I. A. Omar, R. Jayaraman, M. S. Debe, K. Salah, I. Yaqoob, and M. Omar, ‘‘Automating procurement contracts in the healthcare supply chain using blockchain smart contracts,’’ IEEE Access, vol. 9, pp. 37397–37409, 2021, doi: 10.1109/ACCESS.2021.3062471.

[93] R. D. Mendonça, O. S. Gomes, L. F. M. Vieira, M. A. M. Vieira, A. B. Vieira, and J. A. M. Nacif, ‘‘BlockColdChain: Vaccine cold chain blockchain,’’ 2021, arXiv:2104.14357.

[94] S. Saranya, ‘‘Go-Win: COVID-19 vaccine supply chain smart management system using blockchain, IoT and cloud technologies,’’ Turkish J. Comput. Math. Educ., vol. 12, no. 12, pp. 1460–1464, 2021.

[95] P. Zhu, J. Hu, Y. Zhang, and X. Li, ‘‘A blockchain based solution for medication anti-counterfeiting and traceability,’’ IEEE Access, vol. 8, pp. 184256–184272, 2020, doi: 10.1109/ACCESS.2020.3029196.

[96] P. Pandey and R. Litoriya, ‘‘Securing e-health networks from counterfeit medicine penetration using blockchain,’’ Wireless Pers. Commun., vol. 117, no. 1, pp. 7–25, Mar. 2021, doi: 10.1007/S11277-020-07041-7.

[97] N. Saxena, I. Thomas, P. Gope, P. Burnap, and N. Kumar, ‘‘Phar-maCrypt: Blockchain for critical pharmaceutical industry to coun-terfeit drugs,’’ Computer, vol. 53, no. 7, pp. 29–44, Jul. 2020, doi: 10.1109/MC.2020.2989238.

[98] S. Peng, X. Hu, J. Zhang, X. Xie, C. Long, Z. Tian, and H. Jiang, ‘‘An efﬁcient double-layer blockchain method for vaccine production supervision,’’ IEEE Trans. Nanobiosci., vol. 19, no. 3, pp. 579–587, Jul. 2020, doi: 10.1109/TNB.2020.2999637.

[99] C.-L. Chen, Y.-Y. Deng, C.-T. Li, S. Zhu, Y.-J. Chiu, and P.-Z. Chen, ‘‘An IoT-based traceable drug anti-counterfeiting management system,’’ IEEE Access, vol. 8, pp. 224532–224548, 2020, doi: 10.1109/ACCESS.2020.3036832.

[100] M. Debe, K. Salah, R. Jayaraman, and J. Arshad, ‘‘Blockchain-based veriﬁable tracking of resellable returned drugs,’’ IEEE Access, vol. 8, pp. 205848–205862, 2020, doi: 10.1109/ACCESS.2020.3037363.

[101] J. Shi, D. Yi, and J. Kuang, ‘‘Pharmaceutical supply chain management system with integration of IoT and blockchain technology,’’ in Smart Blockchain (Lecture Notes in Computer Science), vol. 11911, M. Qiu, Ed. Cham, Switzerland: Springer, 2019, pp. 97–108, doi: 10.1007/978-3-030-34083-4\_10.

[102] S. Jangir, A. Muzumdar, A. Jaiswal, C. N. Modi, S. Chandel, and C. Vyjayanthi, ‘‘A novel framework for pharmaceutical supply chain management using distributed ledger and smart contracts,’’ in Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2019, pp. 1–7.

[103] A. Kumar, D. Choudhary, M. S. Raju, D. K. Chaudhary, and R. K. Sagar, ‘‘Combating counterfeit drugs: A quantitative analysis on cracking down the fake drug industry by using blockchain technology,’’ in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Conﬂuence), Jan. 2019, pp. 174–178, doi: 10.1109/CONFLUENCE.2019.8776891.

[104] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, ‘‘Where is current research on blockchain technology? A systematic review,’’ PLoS ONE, vol. 11, no. 10, Oct. 2016, Art. no. e0163477.

[105] C. G. Schmidt and S. M. Wagner, ‘‘Blockchain and supply chain relations: A transaction cost theory perspective,’’ J. Purchasing Supply Manage., vol. 25, no. 4, Oct. 2019, Art. no. 100552.

[106] M. E. Messiry and A. E. Messiry, ‘‘Blockchain framework for textile supply chain management,’’ in Proc. Int. Conf. Blockchain. Cham, Switzerland: Springer, 2018, pp. 213–227.

[107] L. Cocco and K. Mannaro, ‘‘Blockchain in agri-food traceability systems: A model proposal for a typical Italian food product,’’ in Proc. IEEE Int. Conf. Softw. Anal., Evol. Reeng. (SANER), Mar. 2021, pp. 669–678.

[108] M. Higginson, M.-C. Nadeau, and K. Rajgopal. (2019). BC’s Occam Problem. [Online]. Available: https://www.mckinsey.com/industries/ﬁnancial-services/ourinsights/BCs-occam-problem

[109] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, ‘‘Design of an anonymity-preserving group formation based authentication protocol in global mobility networks,’’ IEEE Access, vol. 6, pp. 20673–20693, 2018, doi: 10.1109/ACCESS.2018.2827027.