A Comprehensive Analysis of the Research on Blockchain-Enabled Information Sharing Inside a Supply Chain

A Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

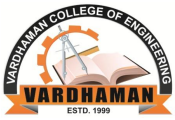
Submitted by

Karthikeya Nagandla	19881A0597
Rayala Suryachand	19881A05A4
Jaithavaram Varshhith Reddy	19881A05B8

SUPERVISOR
Dr.Gouse Baig Mohammad
Associate Professor, CSE

Department of Computer Science and Engineering
VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

March, 2023



VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An Autonomous Institute, Affiliated to JNTUH

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project titled **A Comprehensive Analysis of the Research on Blockchain-Enabled Information Sharing Inside a Supply Chain** is carried out by

Karthikeya Nagandla	**19881A0597**
Rayala Suryachand	**19881A05A4**
Jaithavaram Varshhith Reddy	**19881A05B8**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** during the year 2022-23.

Signature of the Supervisor	**Signature of the HOD**
Dr.Gouse Baig Mohammad	**Dr. Ramesh Karnati**
Associate Professor, CSE	**Professor and Head, CSE**

Project Viva-Voce held on _____

Examiner

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr.Gouse Baig Mohammad**, Associate Professor, CSE and Project Supervisor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. Ramesh Karnati**, the Head of the Department, Department of Computer Science and Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Karthikeya Nagandla

Rayala Suryachand

Jaithavaram Varshhith Reddy

Abstract

A supply chain consists of many stakeholders such as suppliers, carriers and customers. It is often complex due to the rapid development of economic globalization and the intense competition pressure in the market which resulted in information sharing within a supply chain to be fragmented. Blockchain technology can solve this problem by having only a "one trusted ledger" that could reshape the element of data trust. The goal of this paper is to identify and understand the impact of blockchain technology for information sharing within a supply chain. The decentralized nature of blockchain technology offers a high level of transparency and has gained the attention from various sectors to deploy this technology. A systematic literature review in the academic literature was conducted using different databases. Blockchain-enabled information sharing can add value to enhance collaborative work in different types of supply chains such as health and medical, construction and smart city. From our findings, one potential impact of deploying blockchain-enabled information sharing within a supply chain is that it ensures all members in the chain can obtain verified information which enhances collaborative partnerships. Through this in-depth research, we highlighted potential barriers that could impede the development of blockchain technology in supply chain such as the lack of understanding of blockchain technology in businesses and conflict of interests. Future work such as information hiding, in parallel with information sharing, could close the gap in deploying this technology within a supply chain. Understanding the nature of different supply chain is also important to better prepare the deployment of blockchain. We acknowledge that our approach in selecting literatures in our systematic review may exclude certain literatures. Nonetheless, we tried to include as many relevant literatures as possible, to develop a roadmap on the current situation of blockchain-enabled information.

Keywords: lock chain, smart contract, supply chain management, information sharing.

Table of Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	Description
VCE	Vardhaman College of Engineering
BCT	Blockchain Technology
SHA	Secure Hash Algorithm
IoT	Internet of Things
QoS	Quality-of-Services
P2P	Peer-to-Peer
DRY	Don't Repeat Yourself principle
MVC	Model View Controller
AVD	Android Virtual Device
IDE	Integrated Development Environment
UML	Unified Modeling Language
MVT	Model-View-Template
QR Code	Quick Response Code
API	Application Programming Interface
CSRF	Cross Site Request Forgery
SME	Small and Medium-sized Enterprises

CHAPTER 1

Introduction

A supply chain comprises many stakeholders such as suppliers, manufacturers, retailers, carriers and customers [1]. With the rapid development of economic globalization and the intense competition pressure in the market, supply chains have become highly complicated and dynamic [2]–[3]. This is mainly due to the fact that customers are now more demanding, expecting better customized products and better customer service that comes with an acceptable speed and cost. In order to adapt efficiently to the changes in the market and remain competitive, companies are now focusing on their core function, and are moving towards a collective and collaborative effort [3] such as outsourcing, development of advanced value chains, and open innovation [4]. Consequently, the numbers of members within a supply chain have increased rapidly. These members are often scattered globally, which result in information to be highly fragmented. Thus, to better manage and facilitate information sharing among the members within a complex supply chain, a higher co-ordination cost is needed in the form of e.g. quality systems, production standards, etc. However, information asymmetry still exists in the current supply chain.

Information asymmetry is often connected to opportunism in transaction cost economics [5], where information is not fully shared among collaborative partners, and creates room for dishonesty between partners. This interpretation is not what this paper deals with. Our understanding of information asymmetry is defined as hidden information [6], [7], which could be either intentional or unintentional. This happens when not all members within a supply chain obtain product information equally and indiscriminately [8]. Typically, product manufacturers have advantages in controlling and hiding the volume, accuracy and types of information to share with other members, and even to consumers [8], [9]. Conflict of interest is one of the main reasons for hiding information

from other members [10]. This low transparency and highly controlled information flow reduce data trust among members and the efficiency of a supply chain. Data trust in this context refers to the reliability of information and data provided by trade partners within a supply chain or central authority [11]. Accurate data trust in information sharing can act as a catalyst to improve the efficiency in a supply chain.

To decrease the transaction cost, information sharing is identified as an important strategy. Information sharing is where members within a supply chain share information such as product specification, the state of product, ownership, location of data, and even the environmental impact [11]. Information sharing is important for firms which go beyond decision making processes such as increasing profit margin and logistics planning. It is also a key element to enhance collaborative work among members. However, information is constantly transforming from the beginning to the end of the supply chain [10] and the volume of information increases exponentially. With the large volume of information distributed, this could confuse firms and buyers on which data to trust [8], because there is no verification of the truthfulness of the information provided. Thus, there is a need for a better information sharing tool to combat fraud, pilferage, and enhance poorly performing supply chains.

Blockchain technology can solve this problem by having only a "one trusted ledger" that could reshape the element of trust. It is a type distributed ledger technology that can be a solution to a trustable information sharing, by providing a permanent digital footprint to all members in the network. This means every approved transaction occurred throughout the supply chain is recorded in a tamper-evidence environment. Any malicious attempt to alter the information will be obvious and evidential. Blockchain technology can also couple with the Internet of things (IoT) and smart devices to digitize and automate processes to collect and share information in real-time with other members, which improves the transparency and increases the efficiency of a supply chain. These potential impacts on supply chain has caught the attention of many researchers. However, the holistic contribution and barriers of

blockchain-enabled information sharing within a supply chain remain unclear. Therefore, in this paper, we will investigate and understand how blockchain technology can change current information sharing within a supply chain.

1.1 Motivation

There are two main motivations for this project:

1.Blockchain technology can solve this problem by having only a "one trusted ledger" that could reshape the element of trust.

2.Blockchain technology can also couple with the Internet of things (IoT) and smart devices to digitize and automate processes to collect and share information in real-time with other members, which improves the transparency and increases the efficiency of a supply chain.

1.2 Problem Definition

A supply chain comprises many stakeholders such as suppliers, manufacturers, retailers, carriers and customers [1]. With the rapid development of economic globalization and the intense competition pressure in the market, supply chains have become highly complicated and dynamic [2]–[3]. This is mainly due to the fact that customers are now more demanding, expecting better customized products and better customer service that comes with an acceptable speed and cost. In order to adapt efficiently to the changes in the market and remain competitive, companies are now focusing on their core function, and are moving towards a collective and collaborative effort [3] such as outsourcing, development of advanced value chains, and open innovation [4]. Consequently, the numbers of members within a supply chain have increased rapidly. These members are often scattered globally, which result in information to be highly fragmented.

1.3 Objective of Project

There are two main objectives for this project:

1.Blockchain is a type distributed ledger technology that can be a solution to a trustable information sharing, by providing a permanent digital footprint to all members in the network..

2.Reaching a consensus agreement by all participants in the network, before recording it permanently, is the key feature of blockchain technology [2], [11].

Blockchain technology offers a decentralized environment that is built on data trust using a digital approach [2]. Weber et al. [5] insisted that blockchain could be an emerging technology for decentralized and transactional data sharing across a network of untrusted participants. It distributes validated, immutable transactions that are consistent to a large number of members in a network [2], [8]

1.4 Limitations of Project

With the rapid development of economic globalization and the intense competition pressure in the market, supply chains have become highly complicated and dynamic [2]–[3]. It is often complex due to the rapid development of economic globalization and the intense competition pressure in the market .

1.5 Organization of Documentation

Project Introduction, motivation for doing this project, problem definition, objective of project and limitations of project is discussed in chapter 1. Literature survey, existing system, disadvantages of existing system and proposed system in discussed in chapter 2.

Analysis of the project is explained in chapter 3, that starts with introduction followed by software requirement specification that contains user requirements, software requirements and hardware requirements of the project, followed by

content diagrams of project and algorithms and flow chart of the project is discussed in chapter 3. In chapter 4, design of the project is discussed that contains the introduction of the project design, usecase diagram of the project and module design and organization of project. Implementation details of the project is explained in chapter 5, that contains introduction of implementation, explaining the key functions of the project and method of implementation that contains the output screen and result analysis.

In chapter 6, it explains the testing and validation of project with detailed introduction, design of testcases and scenarios of the project followed by project validation details. Conclusion and future enhancements in chapter 7. References are included in the end.

CHAPTER 2

Literature Survey

2.1 Introduction

In the last few years, we have witnessed the potential of Internet of Things to deliver exciting services across several sectors, from social media, business, intelligent transportation and smart cities to the industries [4], [2], [8]. IoT seamlessly interconnects heterogeneous devices with diverse functionalities in the human-centric and machine-centric networks to meet the evolving requirements of the earlier mentioned sectors. Nevertheless, the significant number of connected devices and massive data traffic become the bottleneck in meeting the required Quality-of-Services (QoS) due to the computational, storage, and bandwidth-constrained IoT devices. Mostrecently, the blockchain [3], [4], [5], [6], a paradigm shift, is transforming all the major application areas of IoT by enabling a decentralized environment with anonymous and trustful transactions. Combined with the blockchain technology, IoT systems benefit from the lower operational cost, decentralized resource management, robustness against threats and attacks, and so on. Therefore, the convergence of IoT and blockchain technology aims to overcome the significant challenges of realizing the IoT platform in the near future. Blockchain, a distributed append-only public ledger technology, was initially intended for the cryptocurrencies, e.g., Bitcoin1 . In 2008, Satoshi Nakamato [7] introduced the concept of blockchain that has attracted much attention over the past years as an emerging peer-to-peer (P2P) technology for distributed computing and decentralized data sharing. Due to the adoption of cryptography technology and without a centralized control actor or a centralized data storage, the blockchain can avoid the attacks that want to take control over the system. Later, in 2013, Ethereum, a transaction-based state-machine, was presented to program the blockchain technologies. Interestingly, due to its unique and attractive features such as:

transactional privacy, security, the immutability of data, auditability, integrity, authorization, system transparency, and fault tolerance, blockchain is being applied in several sectors beyond the cryptocurrencies. Some of the areas are identity management [9], intelligent transportation [10], [11] , supply-chain management, mobile-crowd sensing [6], agriculture [7], Industry 4.0 [8], [9], Internet of energy [2], and security in mission critical systems [3]. As shown in Fig. 1, the blockchain structure is composed of a sequence of blocks, which are linked together by their hash values. In the blockchain network, a public ledger maintains the digitally signed transactions of the users in a P2P network. In general, a user has two keys: a public key for other users for the encryption and a private key to read an encrypted message, as shown in Fig. 2. From the blockchain perspective, the private key is used for signing the blockchain transaction and the public key represents the unique address. Asymmetric cryptography is used to decrypt the message encrypted by the corresponding public key. At the initial stage, a user signs a transaction using its private key and broadcasts it to its peers. Once the peers receive the signed transaction, they validate the transaction and disseminate it over the network. All the parties who are involved in the transactions mutually validate the transaction to meet a consensus agreement. Once a distributed consensus is reached, the special node, called as miners, includes the valid transaction into a timestamped block. The block, which is included by the miner, is broadcast back into the network. After validating the broadcast block, which contains the transaction, as well as hash-matching it with the previous block in the blockchain, the broadcast block is appended to the blockchain. Based on the data management and the type of applications, blockchain can classified either as private (permission) or public (permissionless). Both classes are decentralized and provide a certain level of immunity against faulty or malicious users for the ledger. The main differences between private and public blockchains lie in the execution of the consensus protocol, the maintenance of the ledger, and the authorization to join to the P2P network. Detailed examples of these classes are illustrated in [4]. In the context of IoT, blockchains can be classified based on authorization and authentication. As shown in Fig. 3,

in a private blockchain, the centralized trusted authority that manages the authentication and authorization process selects the miners. On the other hand, in a public blockchain (in general, permissionless), there is no intervention of any thirdparty for the miner selection and joining for a new user to the blockchain network. Recently, there is a huge amount of investment from the industries [5], [6] as well as a significant interest from academia to solve major research challenges in blockchain technologies. For example, the consensus protocols are the major building blocks of the blockchain technologies, thus, the threats targeting the consensus protocols become a significant research issue in the blockchain. Furthermore, blockchain forks bring threats to the blockchain consensus protocols. Moreover, it is observed that the vulnerability is about 51 percent for a new blockchain [27]. At the same time, maintenance of several blockchains requires a significant amount of power consumption [28].

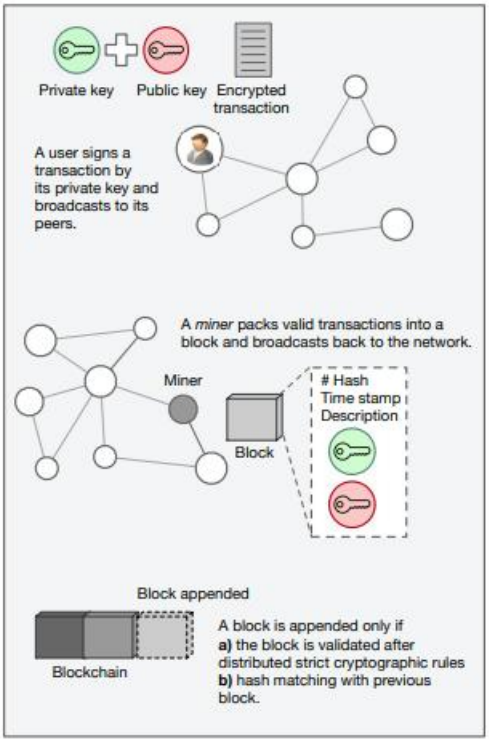


Fig. 2. An illustration of blockchain working methodology.

Figure 2.1: an illusion of blockchain working methodology .

2.2 Existing System

A supply chain comprises many stakeholders such as suppliers, manufacturers, retailers, carriers and customers [1]. With the rapid development of economic globalization and the intense competition pressure in the market, supply chains have become highly complicated and dynamic [2]–[3]. This is mainly due to the fact that customers are now more demanding, expecting better customized products and better customer service that comes with an acceptable speed and cost. In order to adapt efficiently to the changes in the market and remain competitive, companies are now focusing on their core function, and are moving towards a collective and collaborative effort [3] such as outsourcing, development of advanced value chains, and open innovation [4]. Consequently, the numbers of members within a supply chain have increased rapidly. These members are often scattered globally, which result in information to be highly fragmented. Thus, to better manage and facilitate information sharing among the members within a complex supply chain, a higher co-ordination cost is needed.

In supply chain management multiple users such as Suppliers, Consumers may access system and its databases and all supply chain application may allow application owner what data to be shared with customers and what data has to be hided and all this data may store at centralized server or third part cloud servers and some malicious users may tamper or alter that data as this data stored at single centralized server and there will be no other storage to detect this data alterations and to overcome from this problem author is giving brief literature survey on Blockchain technology to migrate supply chain application from centralized server (single server storage) to decentralized Blockchain server (where data will be stored at multiple nodes or server).

2.3 Disadvantages of Existing System

The basic disadvantages include:

1.Data may store at centralized server or third part cloud servers and some

malicious users may tamper or alter that data

2.data stored at single centralized server

3,and there will be no other storage to detect this data alterations

2.4 Proposed System

The goal of this paper is to identify and understand the impact of blockchain technology for information sharing within a supply chain. The decentralized nature of blockchain technology offers a high level of transparency and has gained the attention from various sectors to deploy this technology. A systematic literature review in the academic literature was conducted using different databases. Blockchain-enabled information sharing can add value to enhance collaborative work in different types of supply chains such as health and medical, construction and smart city. From our findings, one potential impact of deploying blockchain-enabled information sharing within a supply chain is that it ensures all members in the chain can obtain verified information which enhances collaborative partnerships. Through this in-depth research, we highlighted potential barriers that could impede the development of blockchain technology in supply chain such as the lack of understanding of blockchain technology in businesses and conflict of interests. Future work such as information hiding, in parallel with information sharing, could close the gap in deploying this technology within a supply chain. Understanding the nature of different supply chain is also important to better prepare the deployment of blockchain. We acknowledge that our approach in selecting literatures in our systematic review may exclude certain literatures. Nonetheless, we tried to include as many relevant literatures as possible, to develop a roadmap on the current situation of blockchain-enabled information sharing within a supply chain.

To decrease the transaction cost, information sharing is identified as an important strategy. Information sharing is where members within a supply chain share information such as product specification, the state of product, owner-

ship, location of data, and even the environmental impact [11]. Information sharing is important for firms which go beyond decision making processes such as increasing profit margin and logistics planning. It is also a key element to enhance collaborative work among members. However, information is constantly transforming from the beginning to the end of the supply chain [10] and the volume of information increases exponentially. With the large volume of information distributed, this could confuse firms and buyers on which data to trust [8], because there is no verification of the truthfulness of the information provided. Thus, there is a need for a better information sharing tool to combat fraud, pilferage, and enhance poorly performing supply chains [1].

Blockchain technology can solve this problem by having only a "one trusted ledger" that could reshape the element of trust. It is a type distributed ledger technology that can be a solution to a trustable information sharing, by providing a permanent digital footprint to all members in the network. This means every approved transaction occurred throughout the supply chain is recorded in a tamper-evidence environment. Any malicious attempt to alter the information will be obvious and evidential. Blockchain technology can also couple with the Internet of things (IoT) and smart devices to digitize and automate processes to collect and share information in real-time with other members, which improves the transparency and increases the efficiency of a supply chain.

2.5 Advantages of Proposed System

1. Blockchain technology to migrate supply chain application from centralized server (single server storage) to decentralized Blockchain server (where data will be stored at multiple nodes or server).

2. In Blockchain technology same transaction data stored at multiple server with hash code verification and if data alter at one server then it will detected from other server as for same data hash code will get different.

3. Work becomes very speedy.

4. Easy to update information.
5. It contains better storage capacity.
6. It tracks all the information of category, orders etc.

CHAPTER 3

ANALYSIS

3.1 Introduction

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

Complete

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

Versatile

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!
Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.
A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.
Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and click jacking (see Website security for more details of such attacks).

Scalable

Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

3.2 Software requirement specification

The concept of menu management system to be developed is to reduce manual efforts in the restaurant ordering system. It is important that such system is deployed in a real environment and hence the requirements are not abstract and are collected from the real setup.
This chapter has the following structure. The general requirements explain the logical setup of the system and describe the basic functionalities needed for such system. The functional requirements are detailed with the help of use case diagram. They are divided into "Customer", "Waiter" and "Admin" subsections. The system that can be used by any customers without training in multiple setups such as restaurants and bars, has to be developed keeping many non-functional requirements in mind. These non-functional requirements were considered in the very beginning of the project and are explained in details in the following sections.

User requirements

1. Android Phone

2. Internet connectivity

Software requirements

Development:

Operating System: Windows 10

Back end Python 3.8 — Back end Programming language

SQLite— databases

Front end — HTML 5, CSS 3, Bootstrap, JavaScript

Pycharm /Visual Studio[IDE/Editor]

Android Virtual Device(AVD)[for testing or debugging]

Deployment:

Pythoneverywhere.com

Framework:

Django

Hardware Requirements

RAM: A minimum of 4GB is required and it can also be upto16GB. Intel CORE i5 and above.

3.2.1 Algorithms and flow charts

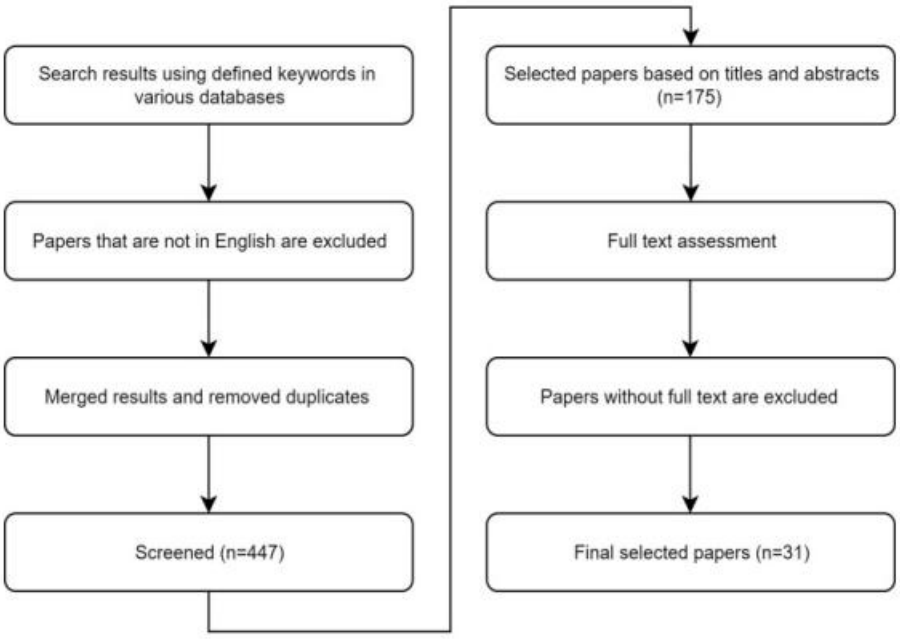


FIGURE 1. Systematic literature review process.

Figure 3.1: Content Diagram of Project

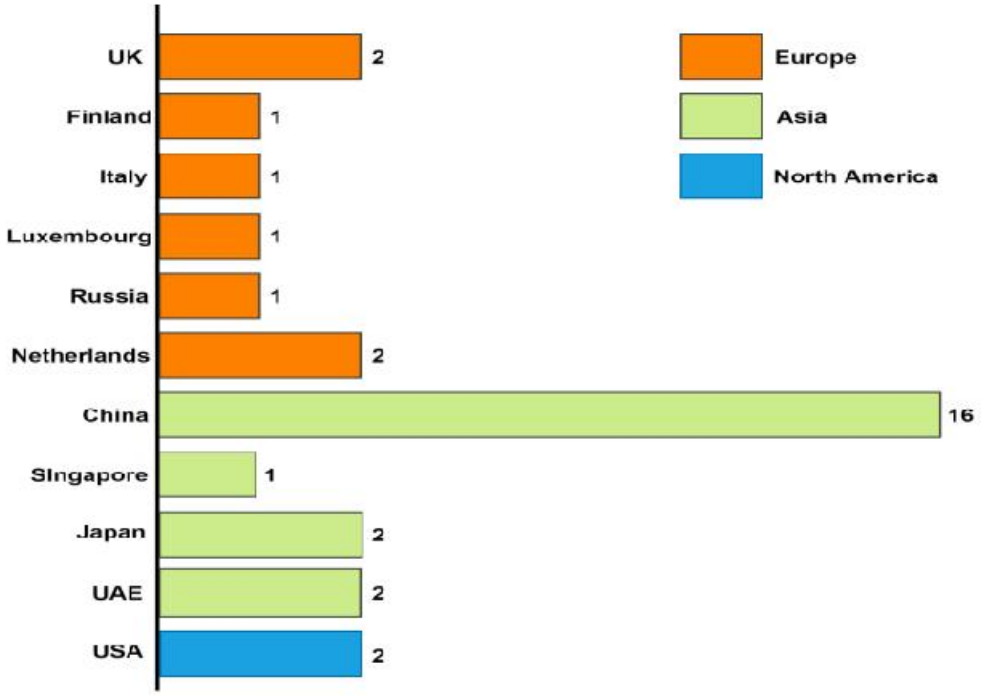


FIGURE 3. **The number of literatures published based on countries.**

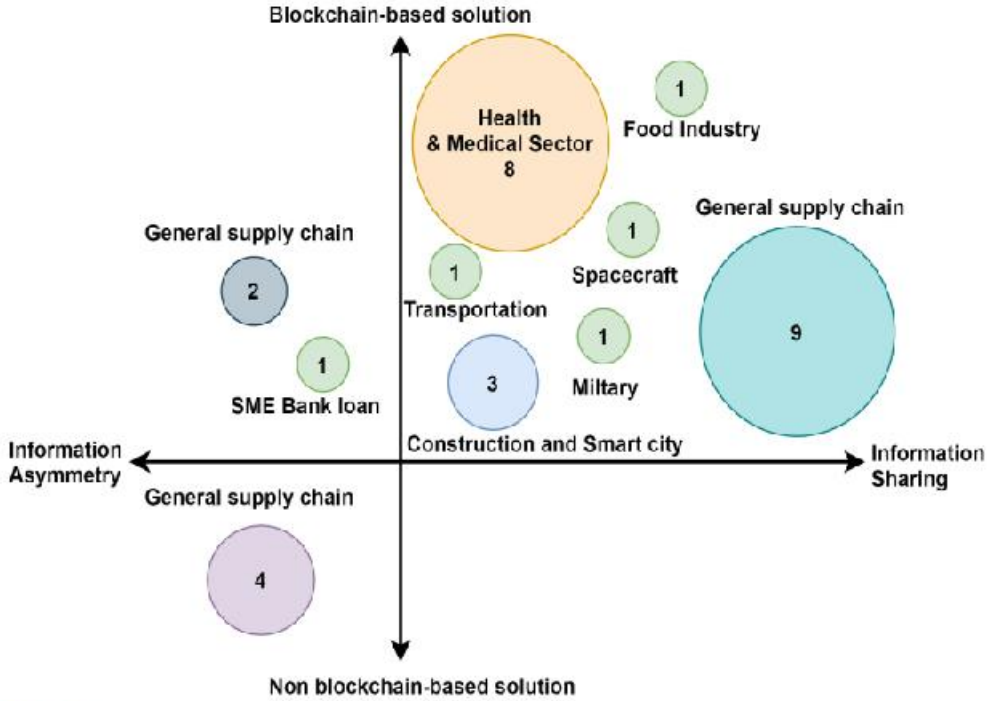


FIGURE 4. **The number of literatures sorted based on the types of supply chain, information and solution.**

CHAPTER 4

DESIGN

4.1 Introduction:

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template. Django web applications typically group the code that handles each of these steps into separate files:

URLs: While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

View: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

Models: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML.

4.2 Module Design AND Organisation:

User Module:

The main aim of the module is to provide all the functionalities related to the users. It tracks all the information and details about the users. We have provided all the signup, read, ordering functionalities to this module. This is a role based module where admin can perform each and every operation on user data but users can only view his/her data. Thus access level restrictions has been implemented on the project. The admin can view the last login time of the users also.

Product Management Module:

The main aim of developing this module is to manage all the details of items being served in the restaurant. So all the records of the items can be update by the admin and the customers can view the food item lists. Also the module manages the order detail i.e the items ordered and the details of the table number, where that food item has to be served. The customers can view the making video of the item they want to order and they can also rate the item.

Confirm Order Module:

The main aim of this module is provide all the functionality related to confirm order. So all confirm order will be managed by admin and customer

will be able to see confirm order. It tracks all the information and details of the confirm order. Admin can Only edit and update the record of the confirm order.

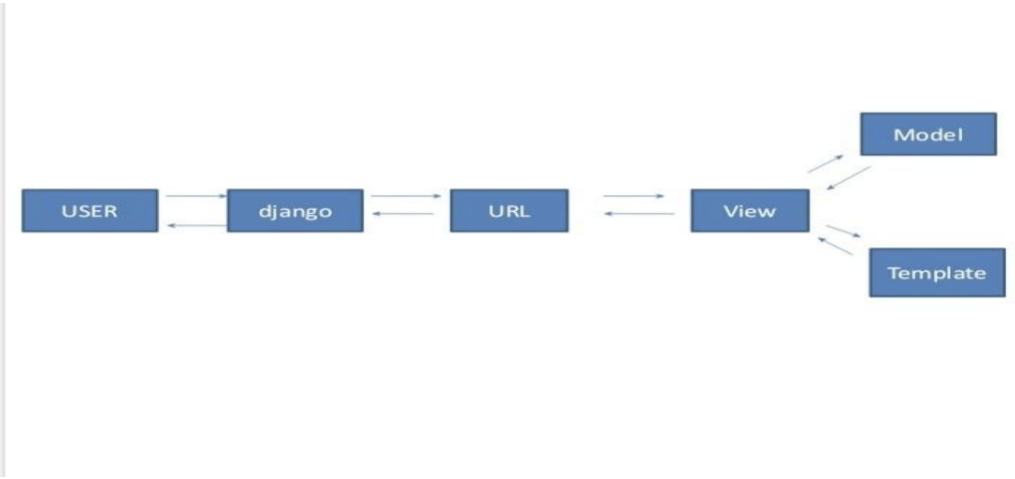


Figure 4.1: Django - MVT Model

4.3 UML Diagram's:

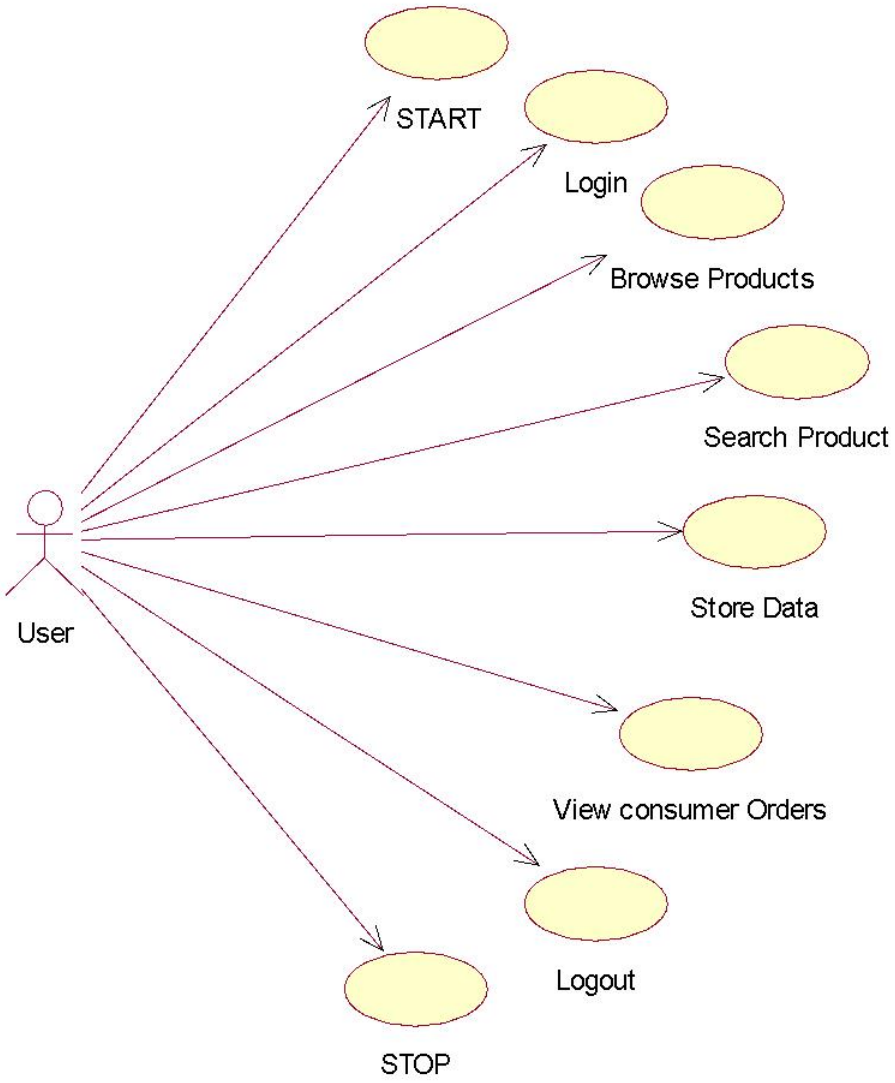


Figure 4.2: Customer Use Case Diagram

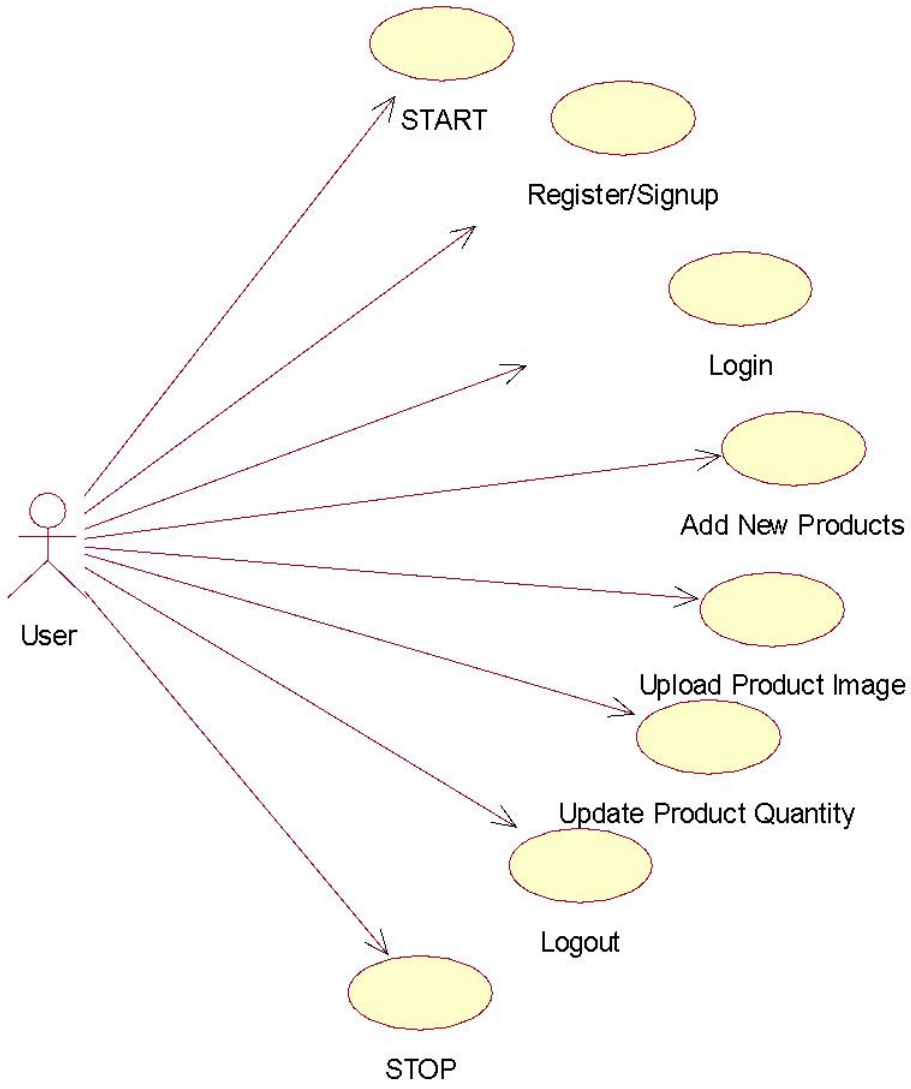


Figure 4.3: Manager Use Case Diagram

CHAPTER 5

IMPLEMENTATION AND RESULT

5.1 Introduction

The customer will be given a QR code at his table. He/she scans the code, it will generate a link which in technical terms is the HTTP request by the client and the Server responds to the request and serves a Web page to the customer at the client side. This page lists the food items available at the restaurant and the customer selects the items and adds them to his cart and then he will be redirected to the payment interface where he pays the bill and this updates the database at the server side and the host receives a notification saying that an order is being received. The host would be able to accept or reject the order as he is provided with an admin control panel. After he accepts the order and the food will be prepared, the customer will receive a notification that his food is ready, either he takes away food or will be served at the table.

5.2 Explanation of Key function

We use Django Forms library to build the Signup and login features. Django provides a Form class which is used to create HTML forms. It describes a form and how it works and appears. It is similar to the ModelForm class that creates a form by using the Model, but it does not require the Model. Each field of the form class map to the HTML form ¡input¿ element and each one is a class itself, it manages form data and performs validation while submitting the form. Django-allauth is an app that provides views, forms, and URLs for registering users, logging them in and out, resetting their passwords, and authenticating users with outside sites like GitHub or Twitter. It supports email-as-username authentication and is extensively documented.

5.3 Method Of Implementation

In our proposed system, QR code Scanning is the first step to access our web page. By simply scanning the provided Qr code. After that, the customers can accesses our web page easily. There is no need to download any application form the play store and also no need to search for our website over the internet. To maintain the uniqueness of the code the QR Code is regenerated every 30 seconds.

In this proposed system, a QR Code is regenerated for every 30 seconds. Customers can scan the QR Code in order to access the website we have developed. The customers can pick their desired Food items from the categorized food item list. And add it into their carts. They can place the orders and pay their bills after confirming their orders. The payment can be done through the Stripe payment or can pay on desk after the satisfactory meal. This can be done according to the customers convenience.

The proposed system is majorly developed using the Django web Framework. The basic layout of the website has been developed using HTML, CSS and bootstrap. Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Django REST framework is a powerful and flexible toolkit for building Web APIs.

5.4 Result Analysis:

In earlier days, industry has lagged behind alternative industries in adopting new technology. But speedy advances in technology and heightened expectations of customers and have forced the industry to bring new innovative methods.

Nowadays, the adoption of wireless technology and emergence of devices has a big role within the industry. The business and services in restaurants are often improved with the mixture of wireless and mobile technologies. The competition in restaurants with regard to business has redoubled with the advancements in ordering techniques.

A digital menu card for restaurants and ordering system using QR code scanning is proposed which is able to streamline the operations to meet customers expectations. The implementation of proposed application uses HTML/CSS and Bootstrap as the front end and at the back end Python, Django Framework along with PostgreSQL database is used.

Using this application customer can order items through scanning QR code easily. After the scanning customer redirect to restaurant menu card thereby customer can select the items and add them to his cart. Then he will be redirected to the payment interface where he pays the bill and this updates the database at the server side and the host receives a notification saying that an order is being received. After that manager accepts the order and the food will be prepared, the customer will receive a notification that his food is ready. This way of communication will happen between the customer.

5.5 Output Screens:

To run project install python 3.7, MYSQL database to store new users details and Blockchain will be used to store product details and order details. Install MYSQL and then copy content from 'DB.txt' file and paste in MYSQL console to create database and then start DJANGO server and open browser and enter URL as 'http://127.0.0.1:8000/index.html' and press enter key to get below screen

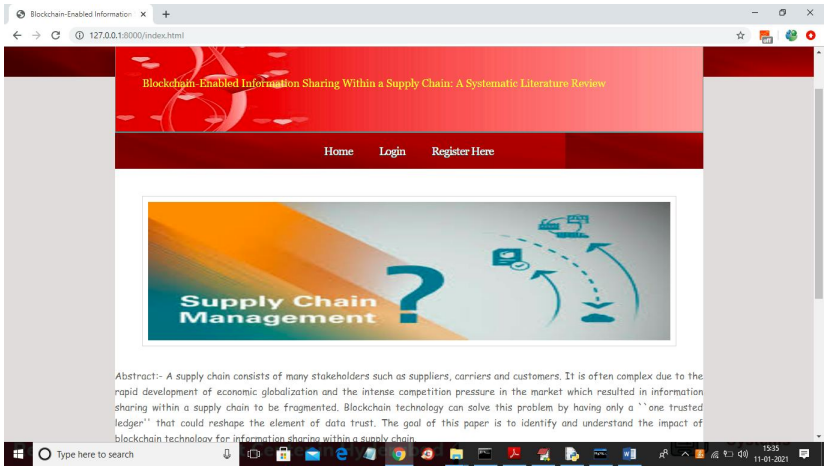


Figure 5.1: In above screen click on 'Register Here' link to get below screen

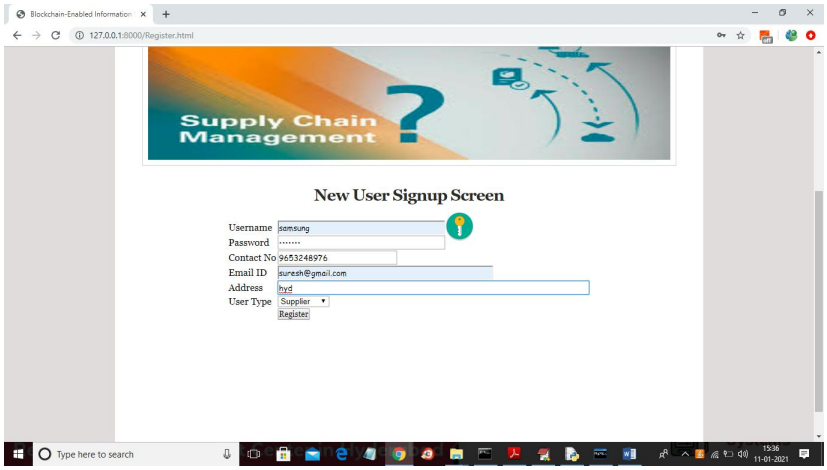


Figure 5.2: In above screen I am adding details of supplier and after clicking on 'Register' button will get below screen

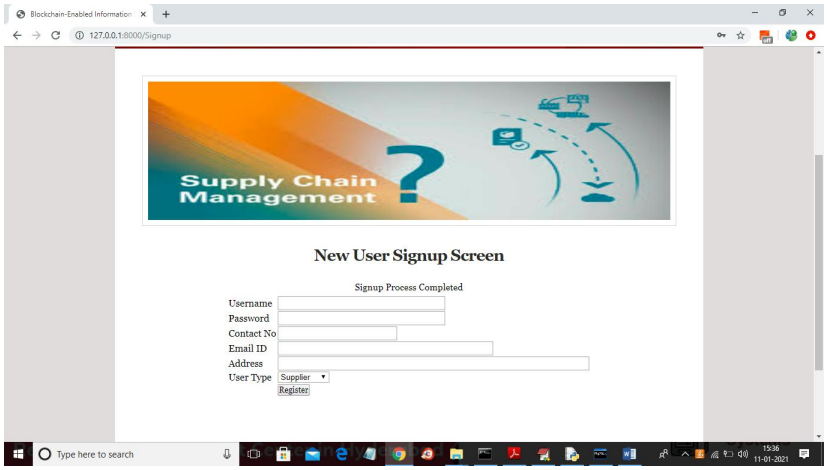


Figure 5.3: In above screen Signup process completed and now add one Consumer

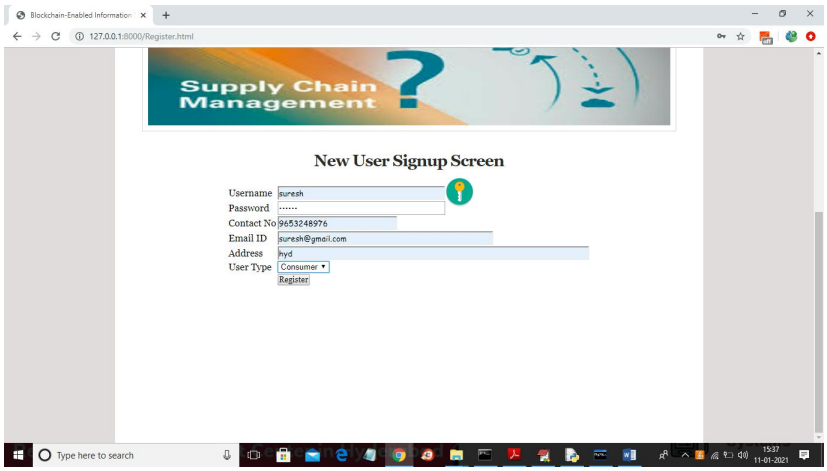


Figure 5.4: In above screen registering one consumer and now click on Login link and then login as supplier

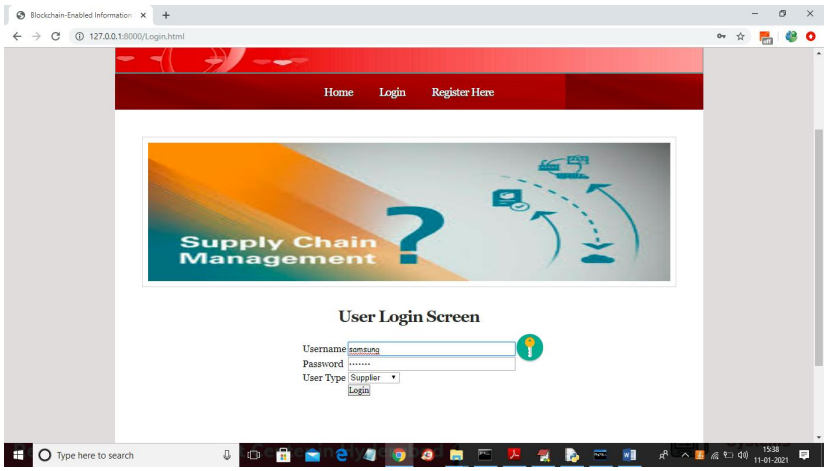


Figure 5.5: In above screen supplier is getting logged in and then after login will get below screen

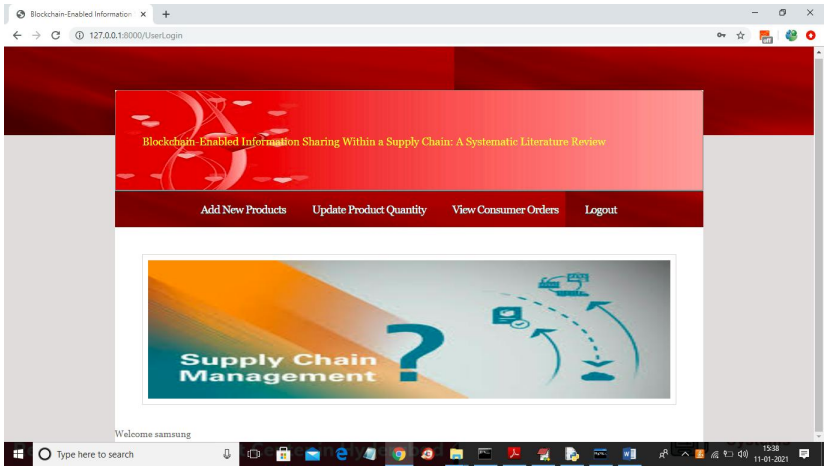


Figure 5.6: In above screen click on 'Add New Products' link to add new transaction to Blockchain

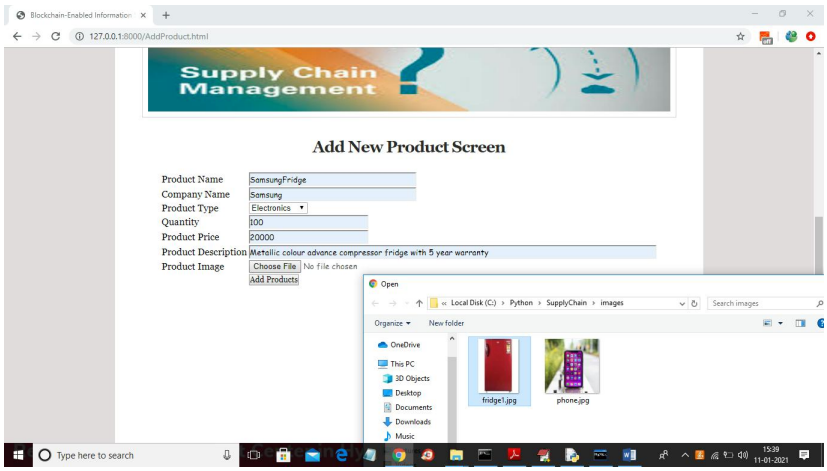


Figure 5.7: In above screen entering new product details and then uploading that product image and then click on 'Add Products' button to add details to Blockchain

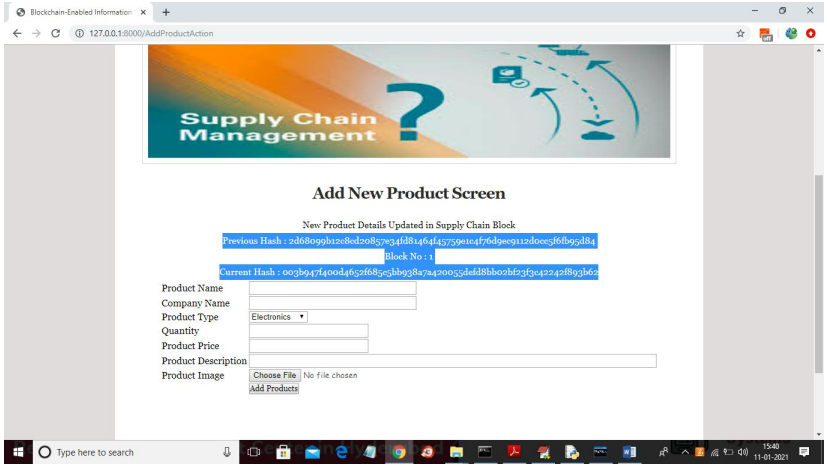


Figure 5.8: In above screen in selected blue colour text we can see to add product details Blockchain has generated old hash code and its block number and its current hash code. Now add another product then we can see the current hash code will match with old hash codeof new product

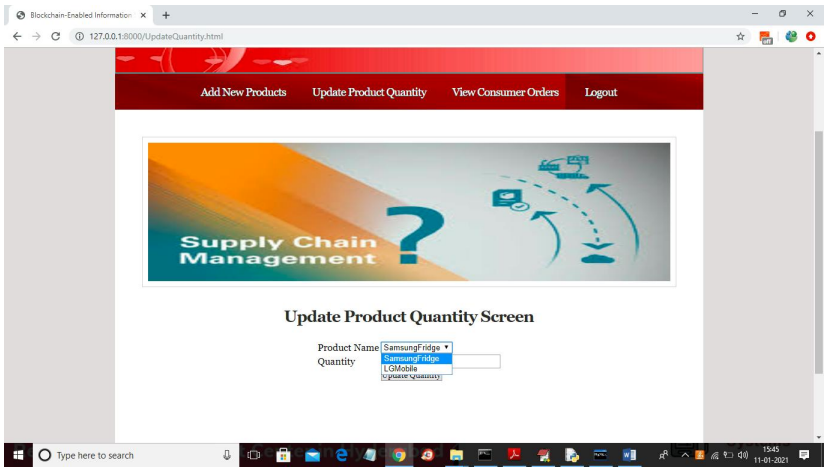


Figure 5.9: In above screen new product details are adding and after clicking on Add Product button will get below screen

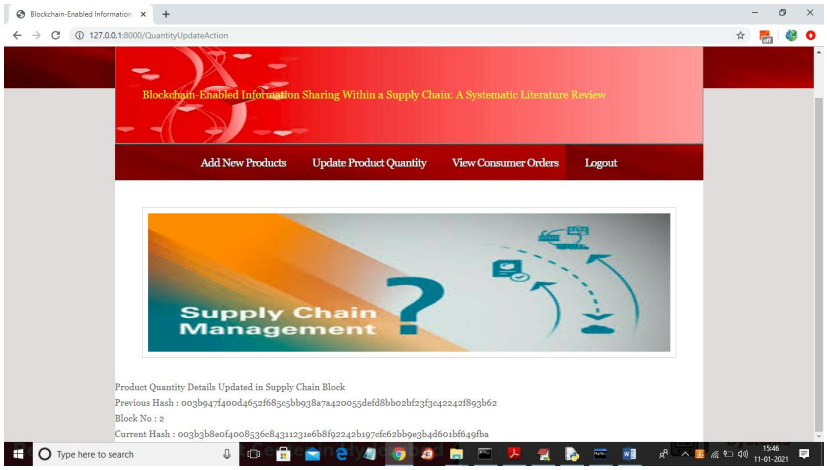


Figure 5.10: In above screen after updating quantity will get Blockchain previous and current hash code and now logout and login as consumer to book product

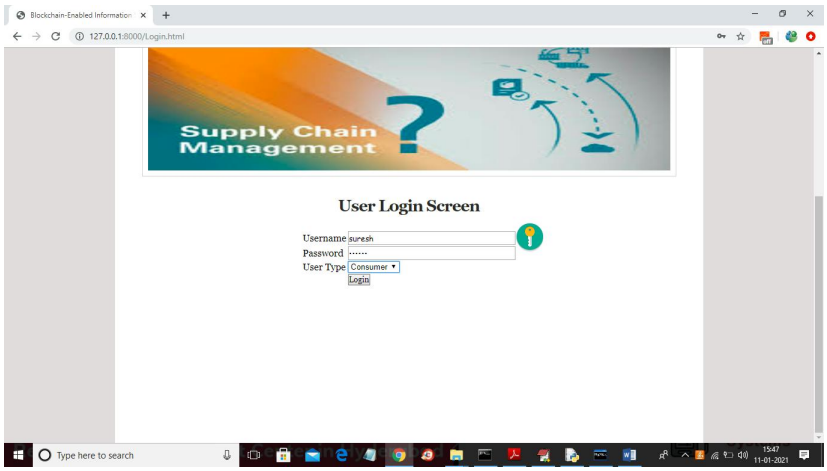


Figure 5.11: In above screen consumer is logged in and after login may get below consumer screen

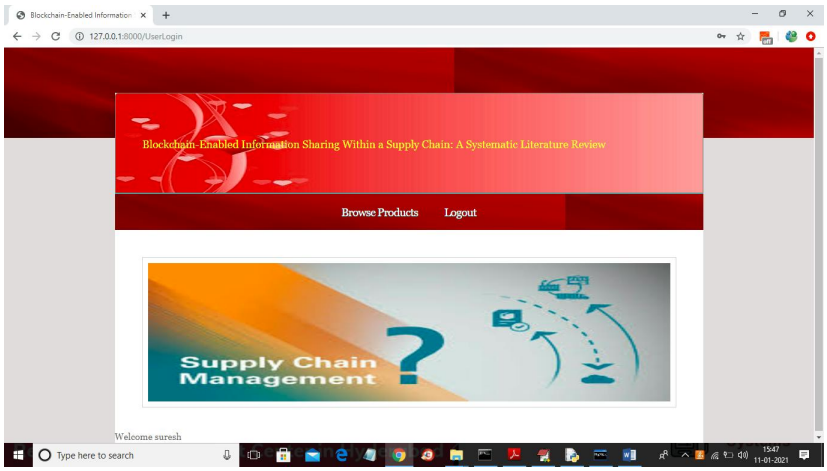


Figure 5.12: In above screen consumer can click on 'Browse Products' link to get below screen

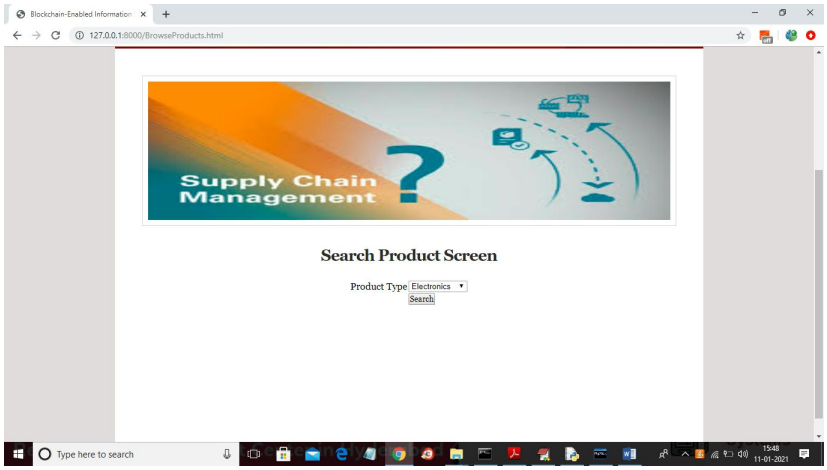


Figure 5.13: In above screen consumer can select desire product type and then click on 'Search' button to get all products of that type in below screen

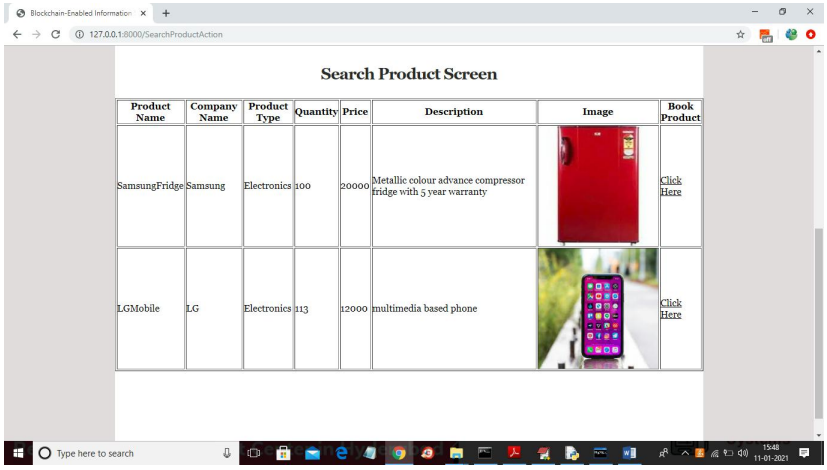


Figure 5.14: In above screen application fetching all product details from Blockchain and displaying to consumer and then consumer can click on 'Click Here' link to booked that product and to get below screen

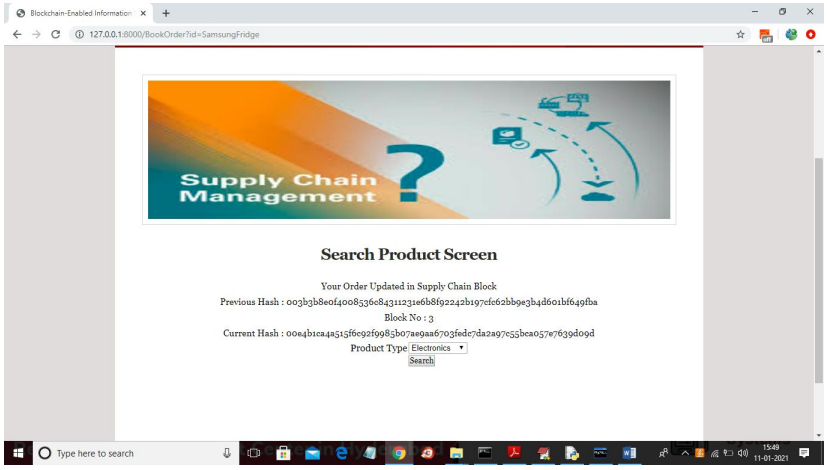


Figure 5.15: In above screen consumer booking data all stored in block chain with block number as 3 with previous and current hash code and now logout and login as supplier to view consumer booking orders

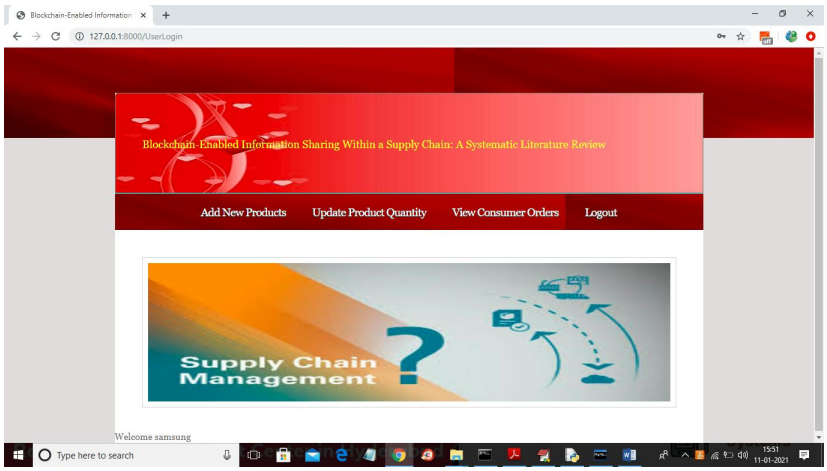


Figure 5.16: In above screen supplier can click on 'View consumer Orders' link to get customer order

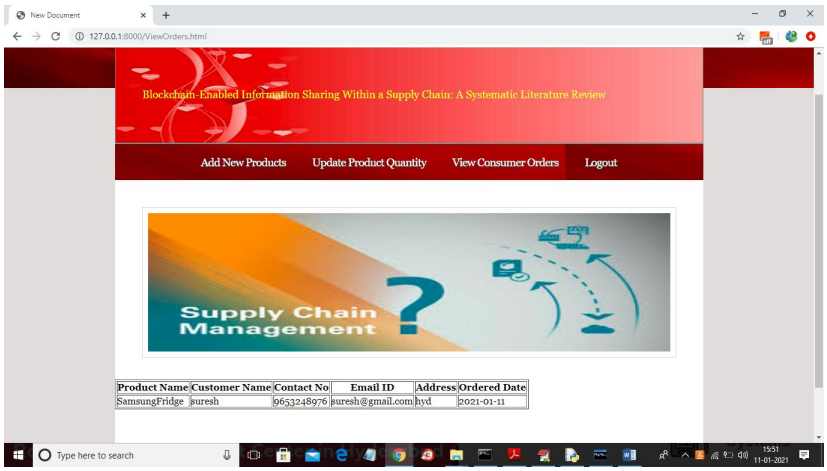


Figure 5.17: In above screen we can see booking ordered from consumer 'suresh'. Similarly all suppliers and consumers may use this application and store their business data on Blockchain decentralized server

CHAPTER 6

TESTING AND VALIDATION

6.1 Introduction

Django provides a test framework with a small hierarchy of classes that build on the Python standard unit test library. Despite the name, this test framework is suitable for both unit and integration tests. The Django framework adds API methods and tools to help test web and Django-specific behavior. These allow you to simulate requests, insert test data, and inspect your application's output. Django also provides an API (LiveServerTestCase) and tools for using different testing frameworks, for example you can integrate with the popular Selenium framework to simulate a user interacting with a live browser.

To write a test you derive from any of the Django (or unittest) test base classes (SimpleTestCase, TransactionTestCase, TestCase, LiveServerTestCase) and then write separate methods to check that specific functionality works as expected (tests use "assert" methods to test that expressions result in True or False values, or that two values are equal, etc.) When you start a test run, the framework executes the chosen test methods in your derived classes.

6.2 Design of test cases and scenarios

For a model, you would first write tests that call each assessor function and make sure that it returns the proper value.
For each function which changes a data field in the model, you would not only test the result of that data field in particular, but you would also test all of the other fields in the model instance to make sure that none of them were modified erroneously.

To restat: if a model has fields a, b, and c, you would create an instance using your constructor, then asset that all three are set properly. Say there's another function, set-a(). You would assert that not only the value of 'a' has changed, but that the values of b and c remain unchanged.
Consider the User model defined, We don't need to explicitly test that first-name and last-name have been stored properly as CharField in the database because that is something defined by Django (though of course in practice you will inevitably test this functionality during development). Nor do you need to test that the date-of-outing has been validated to be a date field, because that is again something implemented in Django.

6.3 Validation

Django provides built-in methods to validate form data automatically. Django forms submit only if it contains CSRF tokens. It uses uses a clean and easy approach to validate data.

The is-valid() method is used to perform validation for each field of the form, it is defined in Django Form class. It returns True if data is valid and place all data into a cleaned-data attribute.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The high volume of information generated from members in a supply chain makes information sharing complicated and highly fragmented. To better facilitate information sharing, different methods, such as engaging a centralized third party, is still commonly used to facilitate information sharing among multiples parties within a supply chain. In addition to that, a legal contract is signed between two institutions or more to enhance collaboration work. However, malicious acts such as alteration to information in centralized database system, without leaving any traces, compromise data integrity and the level of transparency and traceability. Emerging technology such as blockchain technology may transform the current methods of storing information in a decentralization network where no single authority controls over information without compromising data integrity. This paper explored how blockchain-enabled information sharing can rebuild and fortify the element of information and data trust among members within a supply chain.

Blockchain technology has gained great interest within the supply chain due to the decentralized structured with high transparency and traceability. However, research focusing primarily on blockchain-enabled information sharing within a supply chain remains limited. In this systematic literature review, we contributed by identifying what are the significant impacts with the deployment of blockchain-enabled information sharing within a supply chain. The main finding is that this technology ensures members in the chain can obtain verified information which enhances collaborative partnerships. The use of blockchain-based platforms with embedded smart contracts can increase information sharing between trusted and non-trusted institutions with lower security risk. This can benefit any types of supply chain by removing infor-

mation silos which often happen in reality.

Through this in-depth research, we also contributed by highlighting barriers in the development of blockchain technology in supply chains. One of the potential challenges is the unwillingness of firms to share information to other members, due to conflict of interests. The lack of understanding of this technology also impedes the deployment within a supply chain. Till today, many business leaders remain unsure what blockchain is, and how can it contribute to their businesses, while many SMEs claimed they have little knowledge on this. Hopefully these findings can draw a more comprehensive overview to researchers in developing relevant steps to overcome highlighted challenges.

REFERENCES

[1] Quansi Wen, Ying Gao, Zhiling Chen, and Dapeng Wu. "A blockchain-based data sharing scheme in the supply chain by IIoT". In: *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE. 2019, pp. 695–700.

[2] KB Asamatsu. "Preventing malicious acts inexpensive system". In: *Nikkei Business* (2017), pp. 66–68.

[3] Gaurav Tejpal, RK Garg, and Anish Sachdeva. "Trust among supply chain partners: a review". In: *Measuring Business Excellence* (2013).

[4] Henry Chesbrough. *Open business models: How to thrive in the new innovation landscape*. Harvard Business Press, 2006.

[5] William Ouchi. *Markets and Hierarchies: Analysis and Antitrust Implications*. 1977.

[6] Volker Nestle, Florian A Täube, Sven Heidenreich, and Marcel Bogers. "Establishing open innovation culture in cluster initiatives: The role of trust and information asymmetry". In: *Technological Forecasting and Social Change* 146 (2019), pp. 563–572.

[7] Miltiadis Makris. *The Theory of Incentives: The Principal-Agent Model*. 2003.

[8] Yu Cui and Hiroki Idota. "Improving supply chain resilience with establishing a decentralized information sharing mechanism". In: *Proceedings of the 5th Multidisciplinary International Social Networks Conference*. 2018, pp. 1–7.

[9] Anil Singh and James TC Teng. "Enhancing supply chain outcomes through Information Technology and Trust". In: *Computers in human behavior* 54 (2016), pp. 290–300.

[10] Mitsuaki Nakasumi. "Information sharing for supply chain management based on block chain technology". In: *2017 IEEE 19th conference on business informatics (CBI)*. Vol. 1. IEEE. 2017, pp. 140–149.

[11] Yingli Wang, Jeong Hugh Han, and Paul Beynon-Davies. "Understanding blockchain technology for future supply chains: a systematic literature review and research agenda". In: *Supply Chain Management: An International Journal* 24.1 (2019), pp. 62–84.