A

Major Project Report

on

BLOCKCHAIN TECHNOLOGY FOR AGRICULTURAL SUPPLY CHAIN

A Report / dissertation submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology**

In

Computer Science And Engineering

Submitted by

Ch.Thanmai (20EG105609)

A.Devika (20EG105639)

M.Harshith Varma (20EG105712)

N.Sujeeth Kumar (20EG105717)

Under the guidance of

Mr.K.Sadanandam

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENTAKAPUR (V)

TELANGANA

(2023-2024)

DECLARATION

We hereby declare that the Report entitled "**Blockchain technology for agricultural supply chain**" submitted for the award of Bachelor of technology Degree is my original work and the Report has not formed the basis for the award of anydegree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place : Anurag University

Date : 20 April 2024

Ch. Thanmai (20EG105609)

A. Devika (20EG105639)

M. Harshith Varma (20EG105712)

N. Sujeeth Kumar (20EG105717)

CERTIFICATE

This is to certify that the report entitled **“ Blockchain Technology For Agricultural Supply Chain”** that is being submitted by **Ch. Thanmai (20EG105609) , A. Devika (20EG105639) , M. Harshith Varma (20EG105712) , N. Sujeeth Kumar (20EG105717)** in partial fulfilment for the award of B.Tech in **Computer Science and Engineering** to the **Anurag University** is a record of bonafide work carried out by him/her under my guidance and supervision.

The results embodied in this Report have not been submitted to any other University or Institution for the award of any degree or diploma.

Signature of Supervisor	Signature Dean , CSE
Mr. K.Sadanandam	Dr.G.Vishnu Murthy
Assistant Professor	

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mr.K.Sadanandam , Assistant Professor** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical Reviews and constructive comments improved us grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to express our special thanks to **Dr.V.Vijaya Kumar** , Dean School of Engineering, Anurag University, for their encouragement and timely support in our B.Tech program.

We would like acknowledge our sincere gratitude for the support extended by **Dr.G.Vishnu Murthy** , Dean , Dept. of CSE , Anurag University. We also express our deep sense of gratitude to **Dr.V V S S S Balaram** , Acadamic co-ordinator. **Dr. Pallam Ravi**, Project in-charge. **Mr.K.Sadanandam** Project co-ordinator and review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

Ch.Thanmai	(20EG105609)
A.Devika	(20EG105639)
M.Harshith Varma	(20EG105712)
N.Sujeeth Kumar	(20EG105717)

ABSTRACT

In this project we are using IOT network and Blockchain security technology in agriculture food supply chain. In propose work IOT network will be setup in agriculture farms and this IOT will sense food quality growing farms and then report to its nearest cluster head and cluster head will report to base station. Base station will collect food quality data from Cluster Head and then store that data in decentralized Blockchain nodes. This data can be access by various users such as distributors, suppliers, farmers and consumers to know the quality of the food. All existing techniques were using centralized server (single main server) to store data and if this server hack by malicious users, then they can easily alter data on that servers and user's may get wrong or fake data and there is no proper software to detect that alteration and to overcome from this problem Blockchain technology has been introduced. Blockchain support decentralized (data stores at multiple nodes) storage and each node will store data as block of transaction by associating each block with hash code and whenever new data arrive for storage then all nodes will verify hashcode of existing blocks and if all nodes contains same hashcode then data will be consider as secured and unaltered and then new block will be added. If any node report incorrect hashcode then that node considers as attacked and then collect data from genuine nodes. Above verification of hash code is consider as PROOF OF WORK.

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

Our food system travels a complex path from farm to fork. However, this complexity can lead to a lack of transparency and inefficiencies. Blockchain technology is emerging as a game-changer for agriculture supply chains. By creating a secure and shared digital ledger, blockchain can track every step of a product's journey, from seed origin and growing conditions to processing and transport. This increased transparency empowers consumers, builds trust among stakeholders, and paves the way for a more sustainable and efficient agricultural future.

1.1 AIM OF THE PROJECT

Transparency and Traceability: Enable real-time visibility into the entire supply chain, from farm to consumer. Provide a tamper-resistant and immutable ledger that records every transaction and movement of goods. Reduced Fraud and Counterfeiting: Mitigate the risk of fraud by ensuring that the information recorded on the blockchain is secure and cannot be altered. Authenticate the origin of agricultural products, reducing the possibility of counterfeit goods entering the supply chain.

1.2 SCOPE OF THE PROJECT

Supply Chain Mapping: Identify and map the entire agricultural supply chain, including farmers, producers, distributors, retailers, and consumers. Data Collection and Integration: Determine the types of data to be recorded on the blockchain, such as crop information, production methods, transportation details, quality assessments, and more. Integrate data from IoT devices, sensors, and other sources to ensure real-time and accurate information.

1.3 OBJECTIVE OF THE PROJECT

Enhanced Transparency: Increase visibility into the entire agricultural supply chain by providing a transparent and immutable ledger of transactions and movements of agricultural products. Improved Traceability: Enable precise tracking of the origin, production processes, and distribution of agricultural products, reducing the time and effort required for traceability in case of contamination or quality issues.

1.4 PROBLEM STATEMENT

Our project elaborates the major challenges faced by the agriculture industry namely, the food production and food supply chain inefficiency.

1.5 EXISTING SYSTEM

All existing techniques were using centralized server (single main server) to store data and if this server hack by malicious users then they can easily alter data on that servers and user's may get wrong or fake data and there is no proper software to detect that alteration and to overcome from this problem Blockchain technology has been introduced.

Blockchain support decentralized (data stores at multiple nodes) storage and each node will store data as block of transaction by associating each block with hash code and whenever new data arrive for storage then all nodes will verify hash code of existing blocks and if all nodes contains same hash code then data will be consider as secured and unaltered and then new block will be added.

If any node report incorrect hash code then that node consider as attacked and then collect data from genuine nodes. Above verification of hash code is consider as PROOF OF WORK.

1.6 INTODUCTION TO BLOCKCHAIN

Blockchain is a decentralized and distributed ledger technology that enables secure, transparent, and tamper-resistant record-keeping of transactions. It serves as the underlying technology for cryptocurrencies like Bitcoin, but its applications go beyond digital currencies.

Here are some key aspects of blockchain:

- **Decentralization:** Unlike traditional centralized systems, blockchain operates on a network of computers (nodes) that work together to validate and record transactions. This decentralized nature enhances security and eliminates the need for a central authority.
- **Blocks and Chain:** Transactions are grouped together into blocks, and each block contains a reference to the previous block, creating a chain of blocks. This structure

ensures the integrity of the entire transaction history, as altering one block would require changing all subsequent blocks, making tampering highly impractical.

- **Consensus Mechanisms:** Blockchain networks use consensus mechanisms to agree on the validity of transactions and the order in which they are added to the blockchain. Common consensus mechanisms include Proof of Work (used by Bitcoin) and Proof of Stake.
- **Smart Contracts:** These are self-executing contracts with the terms of the agreement directly written into code. Smart contracts automatically enforce and execute the terms when predefined conditions are met. Ethereum is a notable blockchain platform that supports smart contracts.
- **Transparency and Immutability:** Transactions recorded on the blockchain are visible to all participants in the network, promoting transparency. Once a block is added to the blockchain, it becomes extremely difficult to alter, ensuring immutability and security.
- **Cryptocurrencies:** Blockchain's most well-known application is in the creation of digital currencies, such as Bitcoin and Ethereum. These cryptocurrencies use blockchain to enable peer-to-peer transactions without the need for a central authority.
- **Distributed Ledger Technology (DLT):** The term "blockchain" is often used interchangeably with Distributed Ledger Technology (DLT). DLT encompasses a broader range of technologies that distribute and synchronize data across multiple locations, of which blockchain is a specific type.
- **Use Cases:** Beyond cryptocurrencies, blockchain has applications in various industries, including finance (for secure and transparent transactions), supply chain management (for traceability and authenticity), healthcare (for secure and interoperable health records), and more.

Understanding Blockchain Technology:

At its core, a blockchain is a distributed and decentralized ledger that records transactions across a network of computers. Each transaction, or block, is securely linked to the previous one, forming a chain of blocks.

What sets blockchain apart is its tamper-resistant nature; once a block is added to the chain, it becomes nearly impossible to alter the information it contains. This ensures the integrity and transparency of the entire transaction history.

1.7 Internet Of Things (IOT) :

The Internet of Things (IoT) refers to the network of interconnected devices that communicate and exchange data with each other through the internet. These devices, which can range from everyday objects like household appliances and industrial machinery to wearable devices and sensors, are embedded with sensors, software, and other technologies that enable them to collect and exchange data.

In recent years, the agricultural industry has been undergoing a transformative shift, driven by the integration of innovative technologies. One such ground breaking technology poised to revolutionize the agricultural supply chain is blockchain.

Blockchain, originally designed as the underlying technology for cryptocurrencies like Bitcoin, offers a decentralized and transparent platform for recording and verifying transactions. When applied to the agricultural supply chain, it brings about a paradigm shift in how information is shared, transactions are conducted, and trust is established among stakeholders.

1.8 BLOCKCHAIN IN AGRICULTURE :

The Revolution Blockchain technology or blockchain in agriculture can track all types of information about plants, such as seed quality, crop growth, and even the travel of a plant after it leaves the farm. This data can improve supply chain transparency and eliminate concerns associated with illegal and unethical operations.

They can also help track contamination or other issues back to their source in the case of a recall. The major aim of these technologies is sustainability and food security. Consumers can make correct purchasing decisions when they have this amount of openness. They frequently use this information to reward farmers and producers who implement good farming techniques.

How to Apply Blockchain Technology in Agriculture?

There are various blockchain uses in the agricultural sector, and more are being developed based on recent technological advancements. To analyze the key uses of blockchain, it is feasible to divide it into five broad categories:

• Smart Farming Model

• Smart Farming Technology

• Food Supply Chain

• Agriculture Insurance

• Transactions of Agricultural Products

It incorporates information and communication technology (ICT), the internet of things (IoT), multiple sensors, machine learning technologies, and numerous data collecting and analysis equipment such as unmanned aerial vehicles. The connection between smart technology and farming is still new, but with the right security system in place, it can make operations much easier to carry out.

Blockchain helps to maintain data openness and assures that all statistics are completely irreversible. Blockchain's decentralization is by far its greatest strength in smart agriculture. This functionality also facilitates data delivery to the screens of many stakeholders while avoiding data loss and distortion. To ensure transparency, all transactions in a blockchain are time-stamped.

- **Smart Farming Model**

The value and promise of blockchain in agriculture have led to several smart farming models, which help to bring the advantages of this technology together with IoT sensors. One such architecture has been developed for greenhouses, utilizing a private blockchain that can be managed centrally by the farmer.

Another general-purpose approach has been presented, which also uses IoT technologies and blockchain. The basic premise of this framework is to aid in the development of trust among blockchain participants. Numerous stakeholders may use smartphones to access data created at every stage of the farming process, from seeding to product sales.

- **Smart Farming Technology**

Organizations such as Filament have begun to develop smart agricultural technology. One example is a business that sells products with smart farming technology that connects multiple networks to actual items. The business created a coin-sized piece of technology to assist users in safe transactions against a blockchain.

- **Food Supply Chain**

The food supply chain has grown longer and more intensive than ever because of globalization trends. However, there are several challenges in the food supply chain, including food safety, quality, traceability, trust, and supply chain inefficiencies. These factors burden the economy and society and endanger customers' health. Blockchain technology contributes to the resolution of many of these challenges by facilitating the establishment of trust between producers and customers.

Offering specific product information within the blockchain can considerably increase transparency in this process. This has far-reaching repercussions for businesses and farmers alike. It enables businesses to raise the value of their products and hence increase their market competitiveness. It would also make it extremely improbable that providers of low-quality or fraudulent goods would be able to stay in business for very long if their tactics continued.

From a consumer standpoint, the usage of blockchain can be critical in providing people with trustworthy and legitimate information about how their food is produced. It may be used to address a wide range of customer concerns about food quality, safety, and environmental friendliness. Consumers have more flexibility to communicate with food producers as they better grasp their food production process.

When considering the benefits of blockchain from a regulator's perspective, it is evident that this technology provides reliable information to required entities to assist them in enforcing efficient regulations. Many firms have already begun to use blockchain for agriculture in their operations due to the multiple applications of blockchain in the food supply chain and management. Wal-Mart, JD.com, and Alibaba are all using traceability initiatives based on blockchain principles to closely track their whole sales, processing, and food manufacturing process.

- **Agricultural Insurance**

Climate change in recent decades has rendered the entire agricultural process increasingly uncertain. Weather extremes have an impact on agricultural and livestock quality. Farmers often use agricultural insurance systems to mitigate the unpredictability of farming. Farmers can select from various insurance plans that vary in how payouts are calculated and losses are assessed.

A typical type of agricultural insurance, known as indemnity-based insurance, pays farmers according to the conclusions of a professional who examines the farm for damage. However, indemnity-based insurance has several restrictions related to damage estimation and a lack of information from the insurer, which negatively affects farmers and insurance companies.

Blockchain technology enables index-based insurance to give a superior alternative to indemnity-based insurance. It enhances overall insurance process accuracy by triggering a reimbursement based on a quantifiable indicator rather than the loss.

- **Transactions of Agricultural Products**

The use of blockchain technology can significantly speed up the acquisition and sale of agricultural products on ecommerce sites.

It does so in two ways:

Data Security:

Blockchain provides a secure authentication system with private key encryption, increasing the authenticity of all data acquired throughout planting and harvesting.

Supply Chain Management:

In terms of supply chain management, blockchain can increase overall efficiency by lowering the costs associated with signaling. Furthermore, it contributes to safety by providing digital payment solutions that eliminate transaction costs. The use of cryptocurrencies in this technique will also reduce transaction costs.

These changes all contribute to a more trusting relationship between customers and sellers. This has numerous implications for farmers, who stand to make significantly more money on their products and get a larger audience for their goods via the internet.

How Can Blockchain for Supply Chain Help Farmers?

People nowadays want to know exactly where their food originates from. A desire to eat healthy, along with increased acceptance of technology across all disciplines, has led agribusinesses to seek supply chain management software to improve food safety, food quality, and the traceability of the whole farming supply chain.

Precision farming, farmland mapping, IoT sensors, vertical farming systems, location intelligence, crop management software, and transportation technologies enable agricultural businesses to achieve better food production and supply chain management in agriculture. Increased food consumption presents additional concerns, such as counterfeit items jeopardizing food supply networks at various stages.

Farmers and consumers suffer due to a lack of transparency and inefficiency. Finally, blockchain farming and distributed ledger technology (DLT) have the potential to improve agricultural supply chain efficiency, transparency, and trust. By establishing trusting partnerships, blockchain for the agriculture supply chain may empower all market participants.

Agriculture business may be transformed by blockchain for supply chain by:

• Simplifying all phases of the agricultural supply chain

• Following a product from farm to retail shelf

• Increasing food safety and removing counterfeit goods

• Lowering financial risks and encouraging inclusive trade

• Making agricultural finance services available to farmers and companies

• Using Data Science in agriculture to provide smarter market data for improved decision-making; legally demonstrating certifications to necessary authorities

How Blockchain Benefits Agriculture and Food Industry?

Benefits of blockchain in the agriculture supply chain are:

• Transparency

• Analytics

• Security

• Streamlined operations

2. LITERATURE SURVEY

[1] Aiken A. Zooming in on privacy concerns: Video app Zoom is surging in popularity. In our rush to stay connected, we need to make security checks and not reveal more than we think. Index Censorsh. 2020;49(2):24–27.

ZOOM HAS BEEN one of the winners during the Covid-19 lockdowns taking place around the world. The daily number of participants in virtual meetings via the videoconferencing app soared from 10 million in December 2019 to 300 million four months later. But its rapid growth has also sounded alarm bells over privacy, with fears we have rushed to embrace the technology without being fully aware of the risks. "Zoom-bombing" has been well-publicised.

Businesses have had online meetings hacked, while a virtual Holocaust memorial event organised by the Israeli embassy in Germany was hijacked by infiltrators who yelled anti-Semitic slogans and showed photos of Adolf Hitler. But that's not the only issue. The promised end-to-end encryption has yet to materialise; some calls have been routed through China by mistake; data-scraping saw Zoom users' LinkedIn profiles automatically cross-referenced and made public; and users' data has been sent to Facebook.

The company addressed some problems once they were highlighted, but there is little confidence that every flaw has been discovered, and it has been branded "malware" and a "privacy disaster" by security researchers. With Zoom being used for everything from business meetings to family get-togethers and social gatherings, intercepted data can include our most intimate secrets, private conversations, political views and personal beliefs, as well as restricted commercial information. It could prove invaluable to authoritarian regimes, blackmailers, ID hackers and corporate saboteurs.

[2] Bermeo-Almeida O., Cardenas-Rodriguez M., Samaniego-Cobo T., Ferruzola-Gómez E., Cabezas-Cabezas R., Bazán-Vera W. International Conference on Technologies and Innovation, 6-9 November 2018. Guayaquil; Ecuador: 2018. Blockchain in agriculture: a systematic literature review; pp. 44–56.

Blockchain has been used to solve problems from different sectors. In agriculture, Blockchain is being applied for improving food safety, and transaction times. The increasing interest of Blockchain technology in agriculture calls for a clear, systematic overview. In this sense, we present a systematic literature review (SLR) whose objective is to collect all relevant research on Blockchain technology in agriculture to detect current research topics, main contributions, and benefits of applying Blockchain in agriculture.

We have extracted 10 primary studies from scientific databases and web sources published between 2016 and 2018, which means that Blockchain is a recent research area in the agricultural sector. The results show that 60% of papers are focused on food supply chain.

Also, 50% of the studies on Blockchain in Agriculture are dominated by Asian community researchers, especially from China. Similarly, the half of the studies addressed challenges related to privacy and security of the Internet of Things with Blockchain technology.

The goal of this study was to identify research topics, main contributions, and benefits of the blockchain technology in agriculture. We obtained and analyzed 10 primary studies from scientific databases and web sources. The review papers on Blockchain in agriculture is very dominated by Asian research community, especially from China. Only 3 of the 10 reviewed papers are from non-Asian countries.

We attribute this to the fact that agriculture is an important sector in China. In other continents, the concept of blockchain was up to recently adopted. The most frequently addresses research topic is food supply chains. The dominance of studies dealing with food supply chains can be attributed to the importance management for food safety and food quality. Furthermore, 5 of 10 studies present the combination the IoT and blockchain in order to addressed challenges related to privacy and security of the IoT.

As future work, we plan to extend this work by including a wider set of digital libraries such as the Wiley Online Library. Furthermore, we expect this systematic literature review to include more issues and proposed solutions to overcome challenges and limitations of Blockchain technology. Finally, we plan to evaluate the effectiveness of the proposed solutions in an objective way.

[3] Brewin D. The impact of COVID-19 on the grains and oilseeds sector. Can. J. Agric. Econ. /Rev. Can. Agroecon. 2020; 68:185–188.

Brewin (2020) was optimistic about the fate of the Canadian grains and oilseeds sector in 2020 as the COVID-19 pandemic descended on the world. The sector did generate a large crop and, towards the end of 2020, saw a lift in prices. This contributed to record farm income in Canada in 2020. The pace of grain and oilseed exports in Canada and ethanol demand in the east were affected by COVID-19, but the forecast of a “near normal” 2020 was relatively accurate.

Production and prices stayed on track, largely because the world did not impose significant new barriers to trade in cereals and oilseeds and because these sectors have distanced labor in virtually every step of the supply chain which protected these markets from this pandemic. The dominant price factor for the sector remains global demand that had been growing before 2020 relative to the pace of production and may have been stimulated by deficit budgets around the world.

Compared to the tight global stocks, COVID-19 had a minor impact on grain prices which led to steady production worldwide and in Canada. We are still waiting for more evidence to assess the role of federal coordination in the success of the grains and oilseed sector in 2020, but Canada's past participation in trade and safety protocols based on science allowed the grains and oilseed sector in Canada to earn a very good income in 2020.

While the suggestion last spring of "a near normal year" for grains and oilseeds missed significant shifts in ethanol demands and the pace of exports, in the end, 2020 production was near normal and prices were very good after a modest dip in the spring. This lead to forecasted record farm incomes in the Canadian grains and oilseeds sector for 2020. A rise in export demand from China, for biofuels in the EU, and for sanitizers from the health sector in North America, has left current grains and oilseed stocks quite low.

[4] CTA. 2017. Perspectives for ICT and Agribusiness in ACP countries: Start-up financing, 3D printing and blockchain. https://www.cta.int/en/event/perspectives-for-ict-and-agribusiness-in-acp-countries-start-up-financing-3d-printing-and-blockchain-sid002d57e47-75f4-4837-af9b-8fd5d10d5162>

This study aims to investigate blockchain technology for agricultural supply chains during the COVID-19 pandemic. Benefits and solutions are identified for the smooth conduction of agricultural supply chains during COVID-19 using blockchain. This study uses interviews with agricultural companies operating in Pakistan. The findings discover the seven most commonly shared benefits of applying blockchain technology, four major challenges, and promising solutions.

About 100% of the respondents mentioned blockchain as a solution for tracking the shipment during COVID-19, data retrieval and data management, product and transaction frauds, and an Inflexible international supply chain. Roughly 75% of the respondents mentioned the challenge of lack of data retrieval and data management and the Inflexible international supply chain in COVID-19 besides their solutions.

This study can expand existing knowledge related to agricultural supply chains. The experiences shared in this study can serve as lessons for practitioners to adopt the blockchain technology for performing agricultural supply chain during pandemic situations such as COVID-19.

[5] Chofreh A.G., Goni F.A., Jofreh M.G. Enterprise resource planning (ERP) implementation process: project management perspective. Adv. Mater. Res. 2011; 338:152–155.

Enterprise Resource Planning (ERP) system is an advanced manufacturing system that enables the integration of transactions-oriented data and business functions throughout an enterprise. ERP system holds the potential of greatly enhancing organizational performance and establishing competitive advantage. ERP project team can reduce their implementation costs by having a well thought out ERP implementation process. This research examines the ERP implementation process from the project management based perspective.

The theoretical framework is developed to specify the ERP implementation process, which is categorized into five phases in project life cycle: initiating, planning, executing, controlling and closing. The research adopts the iterative triangulation as a methodology to establish this study, and one large company in Iran has been chosen by researchers to examine the proposed framework.

3.PROPOSED METHOD

3.1 ALGORITHM

This method leverages blockchain technology and integrates it with other technologies to create a transparent and efficient agricultural supply chain.

Above Blockchain technology helps in detecting attack nodes and make data secured.

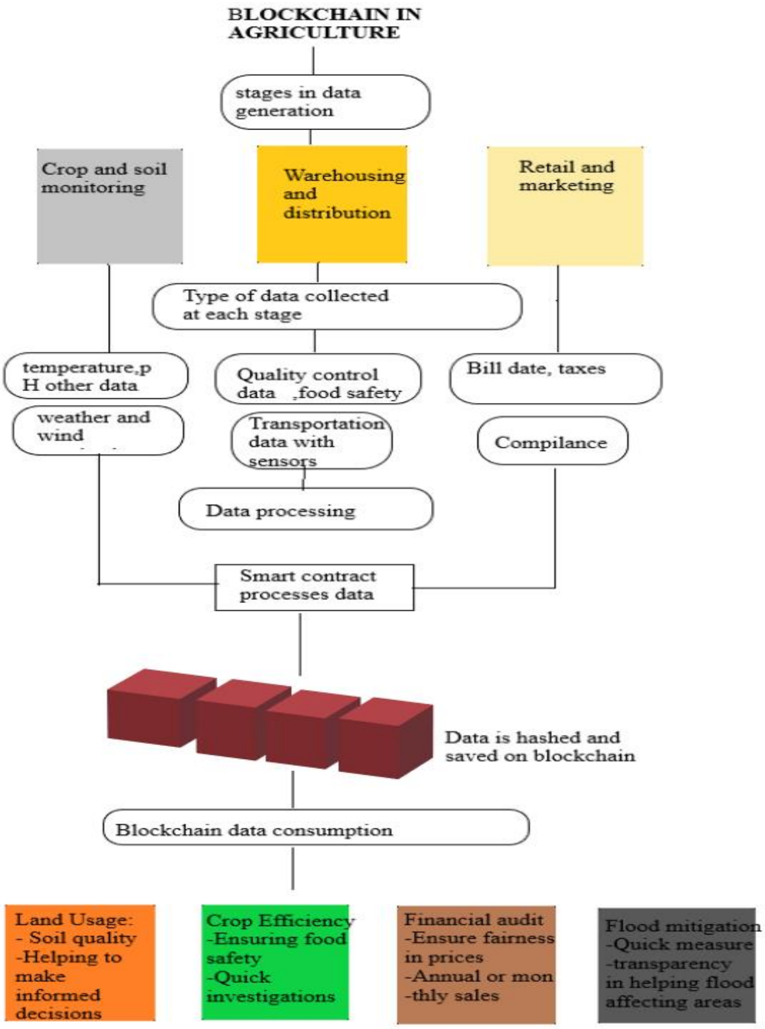


Figure 3.1.1 Working of Blockchain in Agriculture

Generate Network: using this module IOT network will get setup

Cluster Head Selection: all IOT networks exchange there available battery power and then check which IOT covering more number of nodes and can reached to base station with less energy consumption then that node will be elected as cluster head.

Collect Data: using this module IOT will collect/sense food data from agriculture farm.

Data Transmission Routing Phase: using this module IOT will find shortest path to reached cluster head and then transfer data to selected cluster head. CH will send data to base station. Base station will collect data and then store in Blockchain node. Blockchain store each data as block of transaction and will generate hashcode for verification

View Blockchain Data: various users such as consumer, farmers, distributors and many more users may use this module to retrieve data from Blockchain and view it. In this project they have used IOT sensors and agriculture field but we don't have any sensors so we built this concept as simulation.

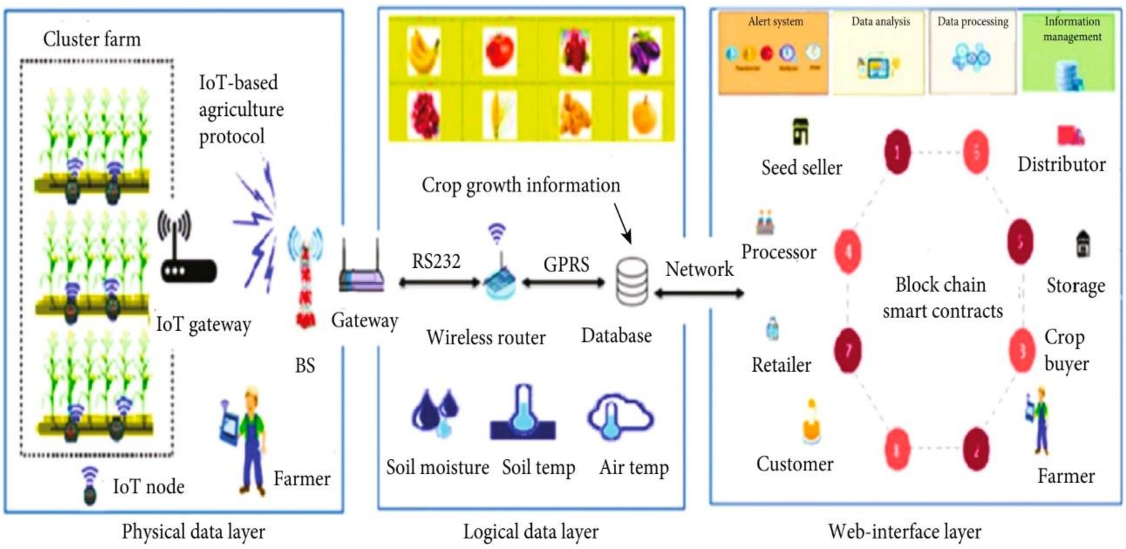


Figure 3.1.2 Layers of Blockchain Architecture

3.2 Block Chain Integration with IoT in the Smart Model

Exclusive blockchain characteristics will combine agricultural and food supply chain processes into a single smart system to ensure that consumers receive healthy food. Figure 7 shows a functional overview of the blockchain

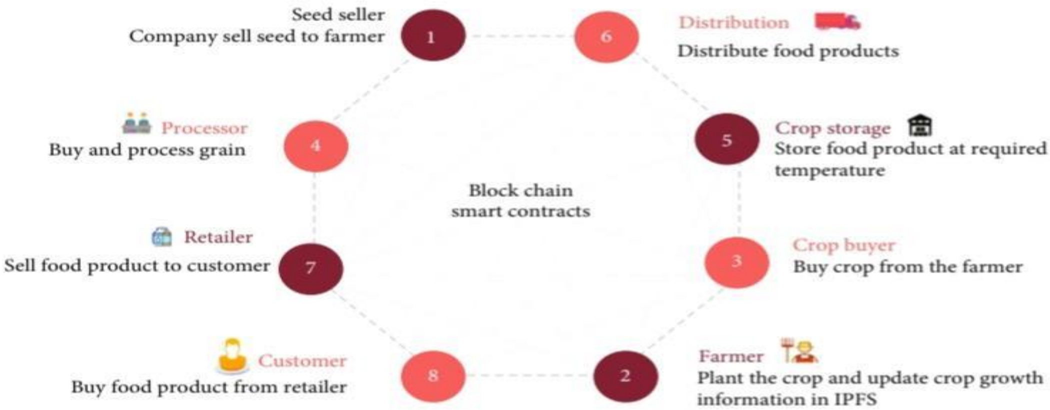


Figure 3.2.1 Blockchain Smart Contracts

The role of stakeholders in the overall system is also discussed. The research used blockchain smart contracts to exchange data between mining nodes in the system. All business transactions are recorded in the shared ledger by mining nodes, and smart contracts receive all transactions in the blockchain in the form of function calls and generate activities, as well as providing access to parties involved in the transaction to exchange control track and receive alerts in the event of a violation. Finally, smart contracts help to maintain the best conditions and respond to food supply chain misappropriations.

3.3 IoT-Based Agriculture Protocol for the Smart Model

IoT nodes are ideal for cluster farms because they consume less energy than WSN and can be further reduced through an efficient clustering protocol. Therefore, this research proposed a new clustering protocol IoT-based agriculture, as shown in Figure 8, based on the LEACH protocol, to reduce energy consumption and extend network life.

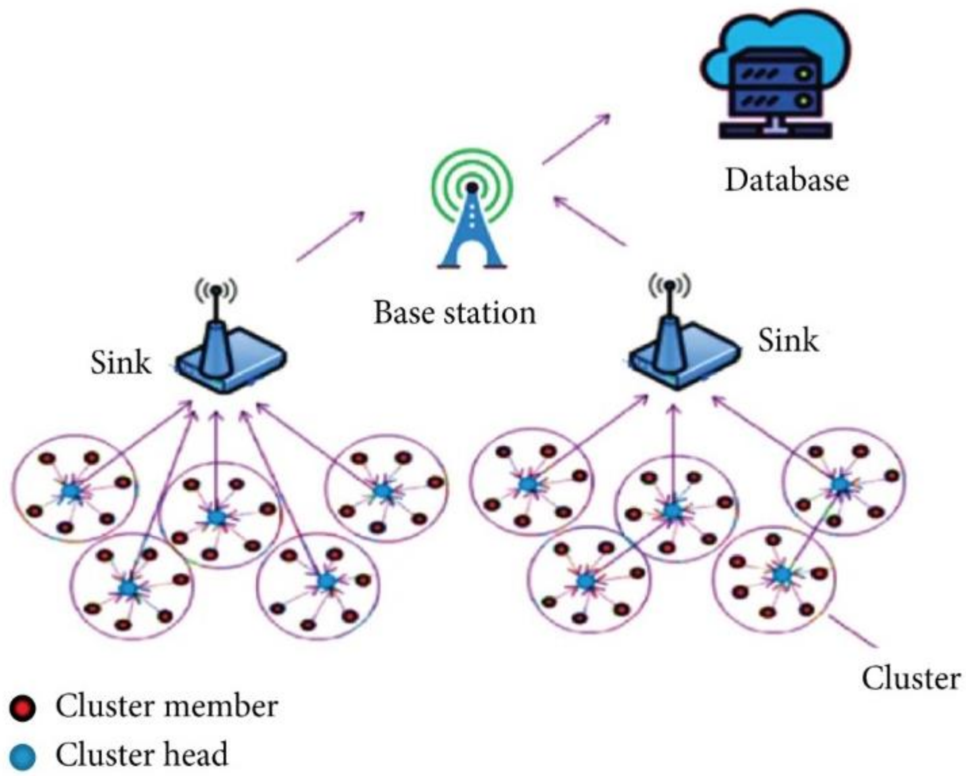


Figure 3.3.1 IoT-Based Agriculture Protocol

3.4 Software and Hardware Requirement

3.4.1 HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15" LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

3.4.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : python
- Tool : Python
- Database : dataset
- Server : Flask

Hardware Interfaces

Intel Core i5 2.00GHz Processor or each and every other processor and 200 GB min RAM 20GB Hard plate, and mouse is required.

Software Interfaces

The Python IDLE is an open-supply web utility that allows you to make and charge records that be essential for stay code, circumstances, portrayals, and story-printed content. Uses encompass realities cleansing and exchange, numerical re-establishment, quantifiable illustrating, realities conviction, framework examining, and divides

4.IMPLEMENTATION

4.1 Modules

A module is a collection of source files and build configurations that allow you to divide your project into many units of functionality. Your project can have one many modules and one module may use another module as a dependency. A module in project open is a high-level description of a functional area, consisting of a group of processes describing the functionality of the module and a group of packages implementing the functionality.
We have Five modules in our project namely:
1.Python IDLE
2.Tkinter
3.Pandas
4.numpy
5.matplotlib

4.2 Module Description

4.2.1 Python IDLE

Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts

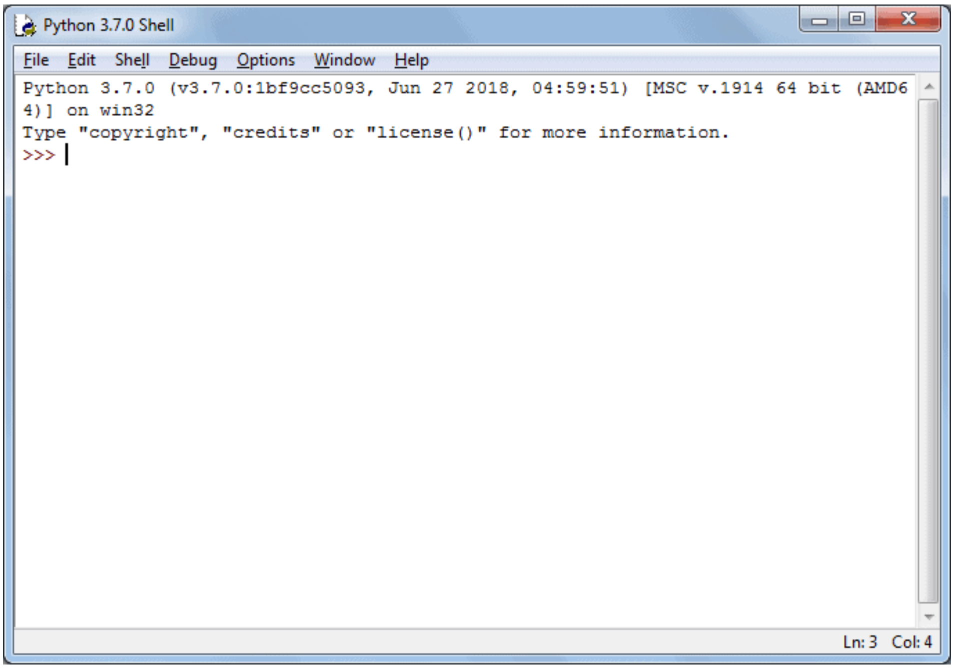


Figure 4.2.1.1 Python Shell

and provides features like syntax highlighting, auto-completion, and smart indent.

IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.

4.2.2 Tkinter

Tkinter is a Python framework that allows users to create graphical user interfaces (GUIs). It is included in all standard Python distributions and is the only framework built into the Python standard library. Tkinter is the most widely used module for GUI applications in Python.

Tkinter provides an interface to the Tk toolkit, which is a collection of graphical control elements, or widgets, for building application interfaces. Tk widgets can be used to construct buttons, menus, data fields, and more. Once created, these graphical elements can interact with features, functionality, methods, data, or other widgets.

Tkinter is cross-platform, so the same code can be used on any operating system, such as Linux, macOS, and Windows. Tkinter is quick and convenient to use, making it a good option for many learners and developers.

4.2.3 Pandas

Pandas is an open-source Python library providing high-performance, easy-to-use data structures and data analysis tools for working with “relational” or “labeled” data. It is generally the first package newcomers to data science in Python learn, as it provides a powerful, flexible, and easy-to-use tool for data manipulation and analysis.

4.2.4 Numpy

NumPy is a Python library that provides a multidimensional array object, various derived objects, and routines for fast operations on arrays. It is an open source project and you can use it freely.

4.3 Sample Code

4.3.1 Iot Stimulation.py

```
import tkinter
from tkinter import *
import math
import random
from threading import Thread
from collections import defaultdict
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
import time
from tkinter import simpledialog
from Block import *
from Blockchain import *
import datetime
import os
global mobile
global labels
global mobile_x
global mobile_y
global text
global canvas
global mobile_list
global filename
global mcen1,mcen2,mcen3
global line1,line2,line3
execution_time = []
option = 0
global root
global crop_growth_details
blockchain = Blockchain()
if os.path.exists('blockchain_contract.txt'):
```

```
    with open('blockchain_contract.txt', 'rb') as fileinput:
        blockchain = pickle.load(fileinput)
    fileinput.close()
def calculateShortestDistance(iot_x,iot_y,x1,y1):
    flag = False
    for i in range(len(iot_x)):
        dist = math.sqrt((iot_x[i] - x1)**2 + (iot_y[i] - y1)**2)  # Calculate the Euclidean
distance between (iot_x[i], iot_y[i]) and (x1, y1)
        if dist < 80:  # dist less than 80
            flag = True
            break
    return flag

 def startDataTransferSimulation(text,canvas,line1,line2,line3,x1,y1,x2,y2,x3,y3):
    class SimulationThread(Thread):
        def __init__(self,text,canvas,line1,line2,line3,x1,y1,x2,y2,x3,y3):
            Thread.__init__(self)
            self.canvas = canvas
            self.line1 = line1
            self.line2 = line2
            self.line3 = line3
            self.x1 = x1
            self.y1 = y1
            self.x2 = x2
            self.y2 = y2
            self.x3 = x3
            self.y3 = y3
            self.text = text

   def run(self):
            time.sleep(1)
            for i in range(0,3):
                self.canvas.delete(self.line1)
                self.canvas.delete(self.line2)
```

```
            self.canvas.delete(self.line3)
            time.sleep(1)
            self.line1 = canvas.create_line(self.x1, self.y1,self.x2, self.y2,fill='black',width=3)
            self.line2 = canvas.create_line(self.x2, self.y2,self.x3, self.y3,fill='black',width=3)
            self.line3 = canvas.create_line(self.x3, self.y3,25, 370,fill='black',width=3)
            time.sleep(1)
        self.canvas.delete(self.line1)
        self.canvas.delete(self.line2)
        self.canvas.delete(self.line3)
        canvas.update()

    newthread = SimulationThread(text,canvas,line1,line2,line3,x1,y1,x2,y2,x3,y3)
    newthread.start()

def generate():
    global mobile
    global labels
    global mobile_x
    global mobile_y
    mobile = []
    mobile_x = []
    mobile_y = []
    labels = []
    canvas.update()
    x = 5
    y = 350
    mobile_x.append(x)
    mobile_y.append(y)
    name = canvas.create_oval(x,y,x+40,y+40, fill="blue")
    lbl = canvas.create_text(x+20,y-10,fill="darkblue",font="Times 7 italic bold",text="Base
Station")
    labels.append(lbl)
    mobile.append(name)
   for i in range(1,20):
```

```
        run = True
        while run == True:
            x = random.randint(100, 450)
            y = random.randint(50, 600)
            flag = calculateShortestDistance(mobile_x,mobile_y,x,y)
            if flag == False:
                mobile_x.append(x)
                mobile_y.append(y)
                run = False
                name = canvas.create_oval(x,y,x+40,y+40, fill="red")
                lbl = canvas.create_text(x+20,y-10,fill="darkblue",font="Times 8 italic
bold",text="IOT "+str(i))
                labels.append(lbl)
                mobile.append(name)

def CHSelection():
    text.delete('1.0', END)
    global mcen1,mcen2,mcen3
    distance = 10000
    for i in range(1,20):
        x1 = mobile_x[i]
        y1 = mobile_y[i]
        energy_consumption = math.sqrt((x1 - 5)**2 + (y1 - 350)**2)
        if energy_consumption < distance and y1 > 5 and y1 < 200:
            distance = energy_consumption
            mcen1 = i
    print(distance)
    distance = 10000
    for i in range(1,20):
        x1 = mobile_x[i]
        y1 = mobile_y[i]
        energy_consumption = math.sqrt((x1 - 5)**2 + (y1 - 350)**2)  #Euclidean distance from
        a base station at (5, 350)
        if energy_consumption < distance and i != mcen1 and y1 > 250 and y1 <= 350 :
```

```
            distance = energy_consumption
            mcen2 = i
    print(distance)
    distance = 10000
    for i in range(1,20):
        x1 = mobile_x[i]
        y1 = mobile_y[i]
        energy_consumption = math.sqrt((x1 - 5)**2 + (y1 - 350)**2)
        if energy_consumption < distance and i != mcen1 and i != mcen2 and y1 > 450 and y1 < 650:
            distance = energy_consumption
            mcen3 = i
    print(distance)
    #calculations to select three cluster heads (mcen1, mcen2, mcen3) based on their proximity to a base station located at coordinates (5, 350) on a canvas. It then updates the appearance of the selected cluster heads on the canvas by changing their color to green and updating their labels.
    text.insert(END,"Selected CH 1 is : "+str(mcen1)+"\n")
    text.insert(END,"Selected CH 2 is : "+str(mcen2)+"\n")
    text.insert(END,"Selected CH 3 is : "+str(mcen3)+"\n")
    canvas.delete(mobile[mcen1])
    canvas.delete(mobile[mcen2])
    canvas.delete(mobile[mcen3])
    canvas.delete(labels[mcen1])
    canvas.delete(labels[mcen2])
    canvas.delete(labels[mcen3])
    # Create green circles for the selected cluster heads
    name = canvas.create_oval(mobile_x[mcen1],mobile_y[mcen1],mobile_x[mcen1]+40,mobile_y[mcen1]+40, fill="green")
    mobile[mcen1] = name
    name = canvas.create_oval(mobile_x[mcen2],mobile_y[mcen2],mobile_x[mcen2]+40,mobile_y[mcen2]+40, fill="green")
```

```
    mobile[mcen2] = name
    name                                                                                    =
canvas.create_oval(mobile_x[mcen3],mobile_y[mcen3],mobile_x[mcen3]+40,mobile_y[mc
en3]+40, fill="green")
    # Create labels for the selected cluster heads
    mobile[mcen3] = name
    lbl=canvas.create_text(mobile_x[mcen1]+20,mobile_y[mcen1]-
10,fill="green",font="Times 10 italic bold",text="CH1-"+str(mcen1))
    labels[mcen1] = lbl
    lbl=canvas.create_text(mobile_x[mcen2]+20,mobile_y[mcen2]-
10,fill="green",font="Times 10 italic bold",text="CH2-"+str(mcen2))
    labels[mcen2] = lbl
    lbl=canvas.create_text(mobile_x[mcen3]+20,mobile_y[mcen3]-
10,fill="green",font="Times 10 italic bold",text="CH3-"+str(mcen3))
    labels[mcen3] = lbl
    canvas.create_oval(50,5,500,245)
    canvas.create_oval(50,240,500,450)
    canvas.create_oval(50,430,500,670)
    canvas.update()

def collectData():
    global crop_growth_details
    crop_growth_details = simpledialog.askstring("Please Enter Crop Growth Details",
   "Please Enter Crop Growth Details")

def dataTansmission():
```

#The dataTansmission function appears to simulate data transmission in a network of IoT devices. It selects a source node (src) and finds the shortest path to a specific destination node (hop). Additionally, it determines a gateway node (gateway) based on the distance to three cluster heads (mcen1, mcen2, mcen3). The function then updates a text widget (text) with information and draws lines on a canvas (canvas) to represent the transmission path.

```
    global crop_growth_details
    global option
    global line1,line2,line3
```

```
text.delete('1.0', END)
src = int(mobile_list.get())
now = datetime.datetime.now()
if option == 1:
    canvas.delete(line1)
    canvas.delete(line2)
    canvas.delete(line3)
    canvas.update()
src_x = mobile_x[src]
src_y = mobile_y[src]
distance = 10000
#If a shortest path (hop) is found, selects a gateway node based on its distance to the three
cluster heads (mcen1, mcen2, mcen3).
hop = 0
gateway = 0
for i in range(1,20):
    temp_x = mobile_x[i]
    temp_y = mobile_y[i]
    if i != src and i != mcen1 and i != mcen2 and i != mcen3 and temp_x < src_x:
        dist = math.sqrt((src_x - temp_x)**2 + (src_y - temp_y)**2)
        if dist < distance:
            distance = dist
            hop = i
if hop != 0:
    hop_x = mobile_x[hop]
    hop_y = mobile_y[hop]
    distance1= math.sqrt((hop_x - mobile_x[mcen1])**2 + (hop_y - mobile_y[mcen1])**2)
    distance2= math.sqrt((hop_x - mobile_x[mcen2])**2 + (hop_y - mobile_y[mcen2])**2)
    distance3= math.sqrt((hop_x - mobile_x[mcen3])**2 + (hop_y - mobile_y[mcen3])**2)
    if distance1 <= distance2 and distance1 <= distance3:
        gateway = mcen1
    elif distance2 <= distance1 and distance2 <= distance3:
        gateway = mcen2
    else:
```

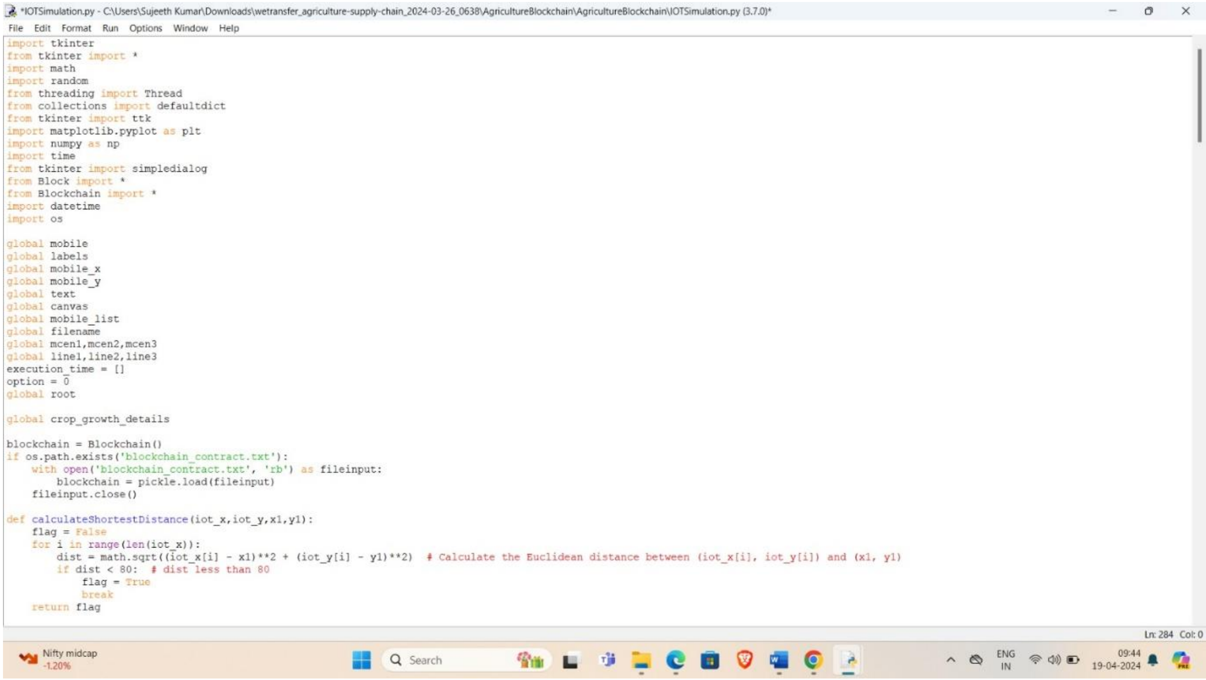
```
        gateway = mcen3
    #If both a gateway and a shortest path are found, updates the text widget with information and draws lines on the canvas.
    if gateway != 0 and hop != 0:
        text.insert(END,"Selected Source Node is : "+str(src)+"\n")
        text.insert(END,"Selected Shortest Path Node with less energy Consumption is : "+str(hop)+"\n")
        text.insert(END,"Selected Nearest CH is : "+str(gateway)+"\n")
        line1=canvas.create_line(mobile_x[src]+20,mobile_y[src]+20,mobile_x[hop]+20,
        mobile_y[hop]+20,fill='black',width=3)
       line2=canvas.create_line(mobile_x[hop]+20,mobile_y[hop]+20,mobile_x[gateway]+20,
       mobile_y[gateway]+20,fill='black',width=3)
  line3=canvas.create_line(mobile_x[gateway]+20,mobile_y[gateway]+20,mobile_x[0]+20,
  mobile_y[0]+20,fill='black',width=3)

startDataTransferSimulation(text,canvas,line1,line2,line3,(mobile_x[src]+20),(mobile_y[src]+20),(mobile_x[hop]+20),(mobile_y[hop]+20),(mobile_x[gateway]+20),(mobile_y[gateway]+20))
        option = 1
        data = "IOT_ID : "+str(src)+"#"+crop_growth_details+"#"+str(now)
        blockchain.add_new_transaction(data)
        hash = blockchain.mine()
        b = blockchain.chain[len(blockchain.chain)-1]
        text.insert(END,"Blockchain Previous Hash : "+str(b.previous_hash)+"\nBlock No : "+str(b.index)+"\nCurrent Hash : "+str(b.hash)+"\n")
        text.insert(END,"Details saved in blockchain\n\n")
        blockchain.save_object(blockchain,'blockchain_contract.txt')
   else:
        text.insert(END,"No shortest path node found. Try another source\n")

def viewData():
#he viewData function appears to retrieve and display data from the blockchain related to a specific IoT device
    text.delete('1.0', END)
```

#The code then iterates through each block in the blockchain, starting from the second block (genesis block is skipped).

```
    src = mobile_list.get()
    for i in range(len(blockchain.chain)):
        if i > 0:
            b = blockchain.chain[i]
            data = b.transactions[0]
            arr = data.split("#")
            temp = arr[0].split(":")
            if temp[1].strip() == src:
                text.insert(END,"IOT ID        : "+temp[1]+"\n")
                text.insert(END,"Crop Details  : "+arr[1]+"\n")
                text.insert(END,"Date Time     : "+arr[2]+"\n")
                text.insert(END,"Blockchain Storage Hashcode     : "+str(b.hash)+"\n\n")


def Main():
    global root
    global tf1
    global text
    global canvas
    global mobile_list
    root = tkinter.Tk()
    root.geometry("1300x1200")
    root.title("BLOCKCHAIN TECHNOLOGY FOR AGRICULTURAL SUPPLY CHAIN")
    root.resizable(True,True)
    font1 = ('times', 12, 'bold')
    canvas = Canvas(root, width = 800, height = 700)
    canvas.pack()

    l1 = Label(root, text='IOT ID:')
    l1.config(font=font1)
    l1.place(x=820,y=10)
```

```
    mid = []
    for i in range(1,20):
        mid.append(str(i))
    mobile_list=ttk.Combobox(root,values=mid,postcommand=lambda:
mobile_list.configure(values=mid))
    mobile_list.place(x=970,y=10)
    mobile_list.current(0)
    mobile_list.config(font=font1)

    createButton = Button(root, text="Generate IOT Network", command=generate)
    createButton.place(x=820,y=60)
    createButton.config(font=font1)

    initButton = Button(root, text="Cluster Head Selection", command=CHSelection)
    initButton.place(x=820,y=110)
    initButton.config(font=font1)

    algButton = Button(root, text="Collect Data", command=collectData)
    algButton.place(x=820,y=160)
    algButton.config(font=font1)

   graphButton=Butto(root,text="Data Transmission Routing Phase",
   command=dataTransmission)
    graphButton.place(x=820,y=210)
    graphButton.config(font=font1)

    exitButton = Button(root, text="View Blockchain Data", command=viewData)
    exitButton.place(x=820,y=260)
    exitButton.config(font=font1)

    text=Text(root,height=20,width=60)
    scroll=Scrollbar(text)
    text.configure(yscrollcommand=scroll.set)
```

```
   text.place(x=820,y=310)
root.mainloop()

if __name__== '__main__' :
   Main ()
```




```
import tkinter
from tkinter import *
import math
import random
from threading import Thread
from collections import defaultdict
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
import time
from tkinter import simpledialog
from Block import *
from Blockchain import *
import datetime
import os

global mobile
global labels
global mobile_x
global mobile_y
global text
global canvas
global mobile_list
global filename
global mcen1,mcen2,mcen3
global line1,line2,line3
execution_time = []
option = 0
global root

global crop_growth_details

blockchain = Blockchain()
if os.path.exists('blockchain_contract.txt'):
    with open('blockchain_contract.txt', 'rb') as fileinput:
        blockchain = pickle.load(fileinput)
    fileinput.close()

def calculateShortestDistance(iot_x,iot_y,x1,y1):
    flag = False
    for i in range(len(iot_x)):
        dist = math.sqrt((iot_x[i] - x1)**2 + (iot_y[i] - y1)**2)  # Calculate the Euclidean distance between (iot_x[i], iot_y[i]) and (x1, y1)
        if dist < 80:  # dist less than 80
            flag = True
            break
    return flag
```

Figure 4.3.1.1 IOT Stimulation.py

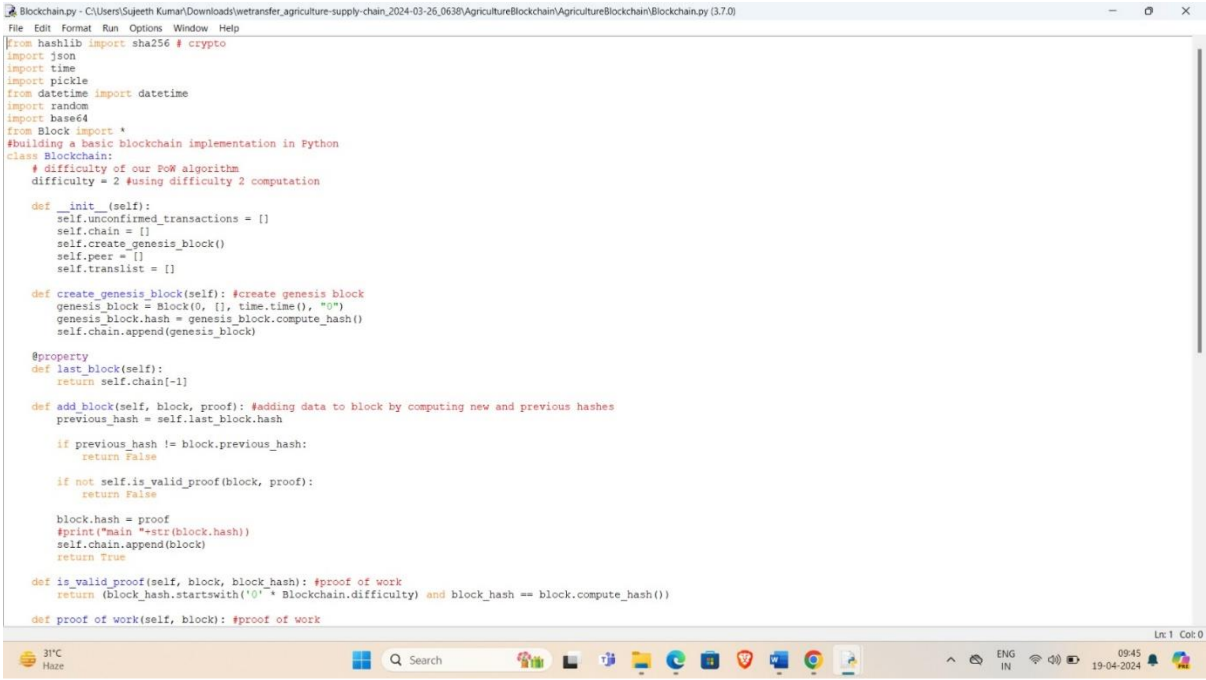
4.3.2 Blockchain.py

```
from hashlib import sha256
import json
import time
import pickle
from datetime import datetime
import random
import base64
from Block import *
#building a basic blockchain implementation in Python
class Blockchain:
```

```
# difficulty of our PoW algorithm
difficulty = 2 #using difficulty 2 computation

def __init__(self):
    self.unconfirmed_transactions = []
    self.chain = []
    self.create_genesis_block()
    self.peer = []
    self.translist = []

def create_genesis_block(self): #create genesis block
    genesis_block = Block(0, [], time.time(), "0")
    genesis_block.hash = genesis_block.compute_hash()
    self.chain.append(genesis_block)

@property
def last_block(self):
    return self.chain[-1]

def add_block(self, block, proof): #adding data to block by computing new and previous
hashes
    previous_hash = self.last_block.hash

    if previous_hash != block.previous_hash:
        return False

    if not self.is_valid_proof(block, proof):
        return False

    block.hash = proof
    #print("main "+str(block.hash))
    self.chain.append(block)
    return True
```

```
    def is_valid_proof(self, block, block_hash): #proof of work
        return (block_hash.startswith('0' * Blockchain.difficulty) and block_hash ==
block.compute_hash())

    def proof_of_work(self, block): #proof of work
        block.nonce = 0
        computed_hash = block.compute_hash()
        while not computed_hash.startswith('0' * Blockchain.difficulty):
            block.nonce += 1
            computed_hash = block.compute_hash()
        return computed_hash

    def add_new_transaction(self, transaction):
        self.unconfirmed_transactions.append(transaction)

    def addPeer(self, peer_details):
        self.peer.append(peer_details)

    def addTransaction(self,trans_details):
#add transaction
        self.translist.append(trans_details)

    def mine(self):
#mine transaction
        if not self.unconfirmed_transactions:
            return False

        last_block = self.last_block

        new_block = Block(index=last_block.index + 1,
                          transactions=self.unconfirmed_transactions,
                          timestamp=time.time(),
                          previous_hash=last_block.hash)
```

```
        proof = self.proof_of_work(new_block)
        self.add_block(new_block, proof)

        self.unconfirmed_transactions = []
        return new_block.index

    def save_object(self,obj, filename):
        with open(filename, 'wb') as output:
            pickle.dump(obj, output, pickle.HIGHEST_PROTOCOL)
```




```
from hashlib import sha256 # crypto
import json
import time
import pickle
from datetime import datetime
import random
import base64
from Block import *
#building a basic blockchain implementation in Python
class Blockchain:
    # difficulty of our PoW algorithm
    difficulty = 2 #using difficulty 2 computation

    def __init__(self):
        self.unconfirmed_transactions = []
        self.chain = []
        self.create_genesis_block()
        self.peer = []
        self.translist = []

    def create_genesis_block(self): #create genesis block
        genesis_block = Block(0, [], time.time(), "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        return self.chain[-1]

    def add_block(self, block, proof): #adding data to block by computing new and previous hashes
        previous_hash = self.last_block.hash

        if previous_hash != block.previous_hash:
            return False

        if not self.is_valid_proof(block, proof):
            return False

        block.hash = proof
        #print("main "+str(block.hash))
        self.chain.append(block)
        return True

    def is_valid_proof(self, block, block_hash): #proof of work
        return (block_hash.startswith('0' * Blockchain.difficulty) and block_hash == block.compute_hash())

    def proof_of_work(self, block): #proof of work
```

Figure 4.3.2.1 Blockchain.py

5.RESULT

Blockchain technology is revolutionizing agriculture supply chains. It creates a secure, shared digital ledger that tracks every step of a product's journey – from farm origin and growing conditions to processing, transport, and reaching store shelves. This newfound transparency empowers consumers to make informed choices, builds trust between participants, and streamlines operations. The result is a more efficient, sustainable, and verifiable food system, ensuring quality and fair practices throughout the agricultural supply chain . In this project they have used IOT sensors and agriculture field but we don't have any sensors so we built this concept as simulation.

5.1 SCREENSHOTS

To run project double click on 'run.bat' file to get below screen

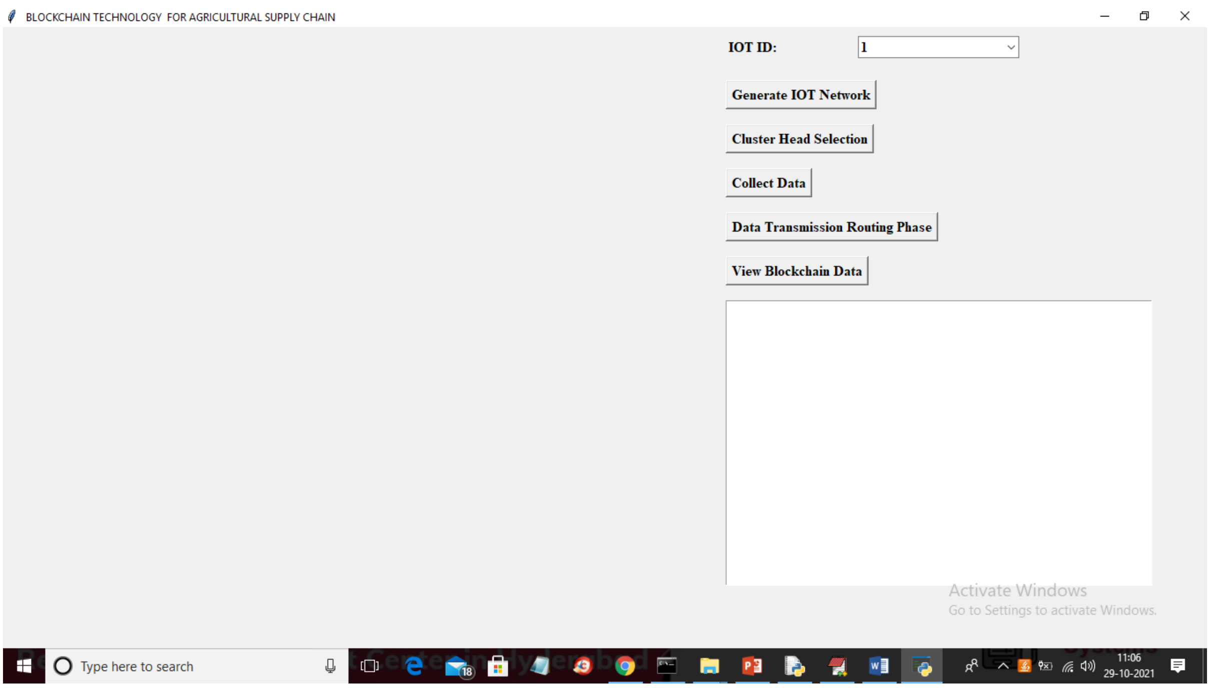


Figure 5.1.1 'run.bat' file

In above screen click on 'Generate IOT Network' button to generate IOT simulation network and to get below screen.

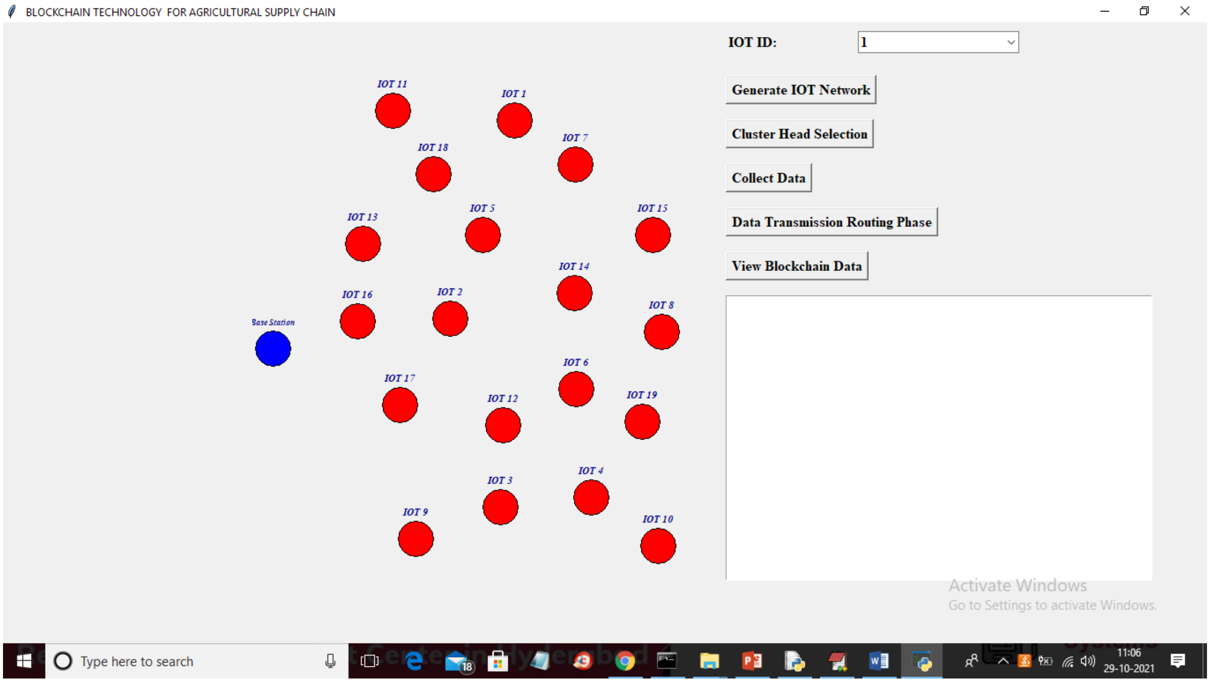


Figure 5.1.2 Generate IOT Network

In above screen each red colour circle represents as IOT sensors installed in agriculture farms and blue colour circle represents Base Station. Now click on 'Cluster Head Selection' button to select cluster head with high available energy and can reach to more IOT with less distance to Base station.

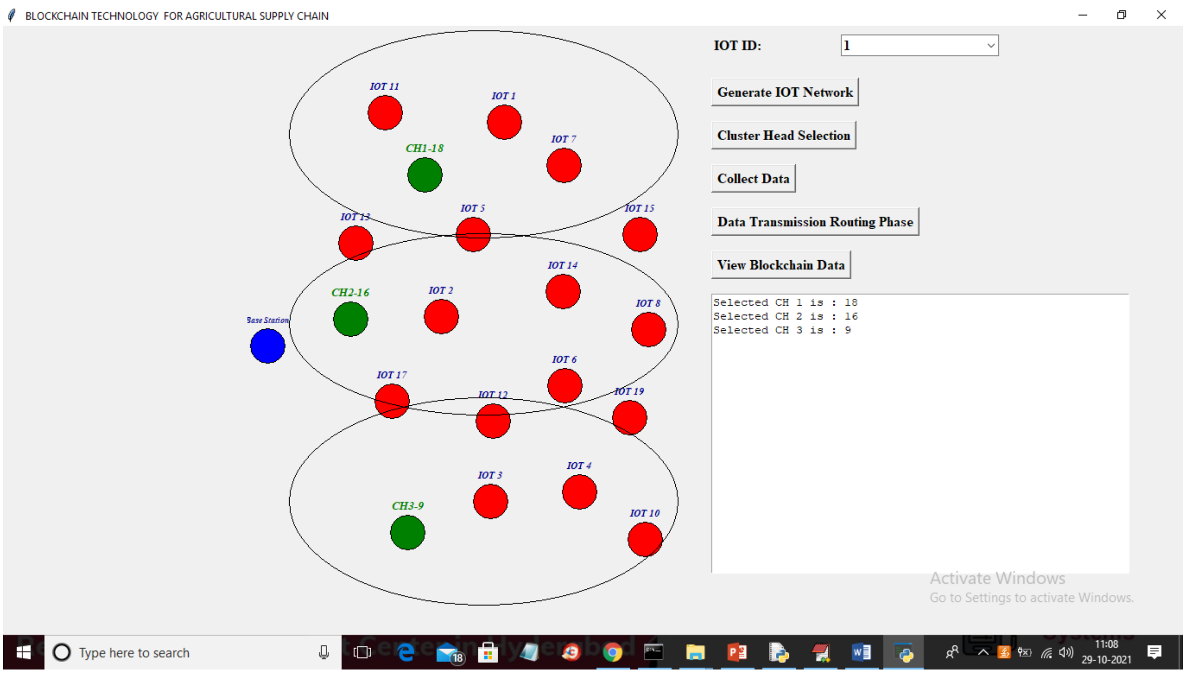


Figure 5.1.3 Cluster Head Selection

In above screen green colour IOT is selected as Cluster Head and all the IOT inside big oval will be consider as cluster member of that cluster and total 3 clusters are generated and now click on ‘Collect Data’ to enter some manual data as we don’t have any sensor to sense data so we collect data from keyboard manually.

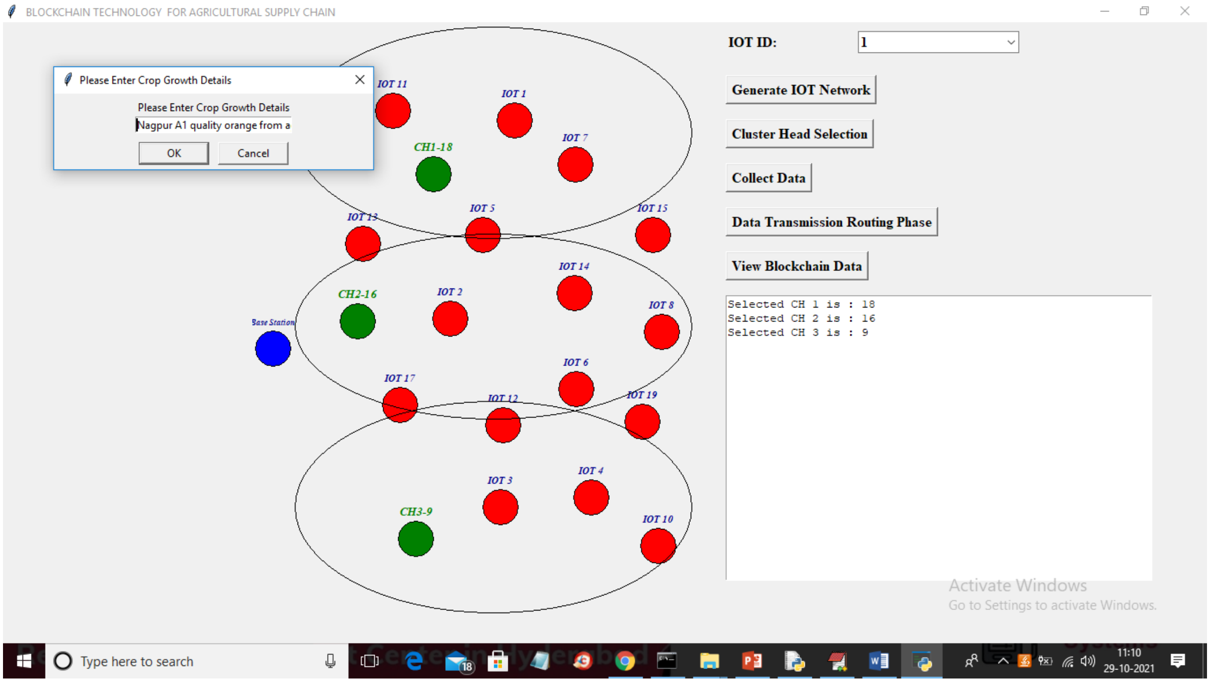


Figure 5.1.4 Collect Data

In above screen as data collection I entered some data and then click on ‘OK’

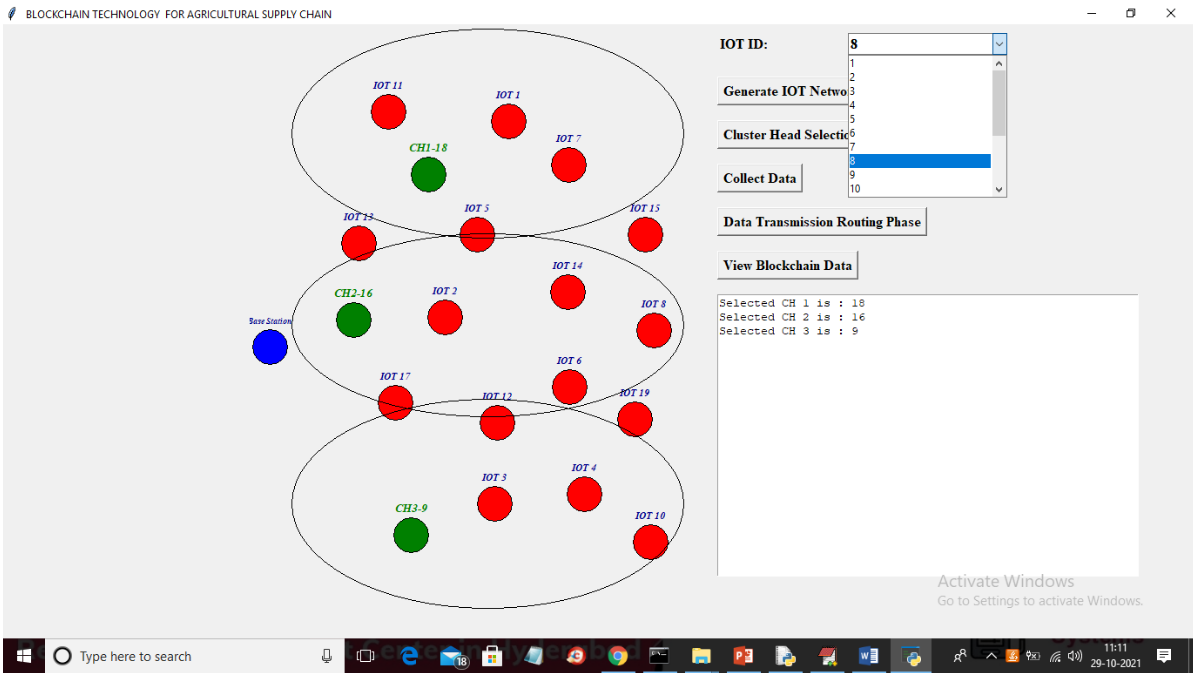


Figure 5.1.5 IOT Data Entering

button to get below screen

In above screen from IOT drop down box I selected sender IOT as 8 and then click on ‘Data Transmission Routing Phase’ button to allow IOT 8 to select shortest path to reached cluster head and then transfer data.

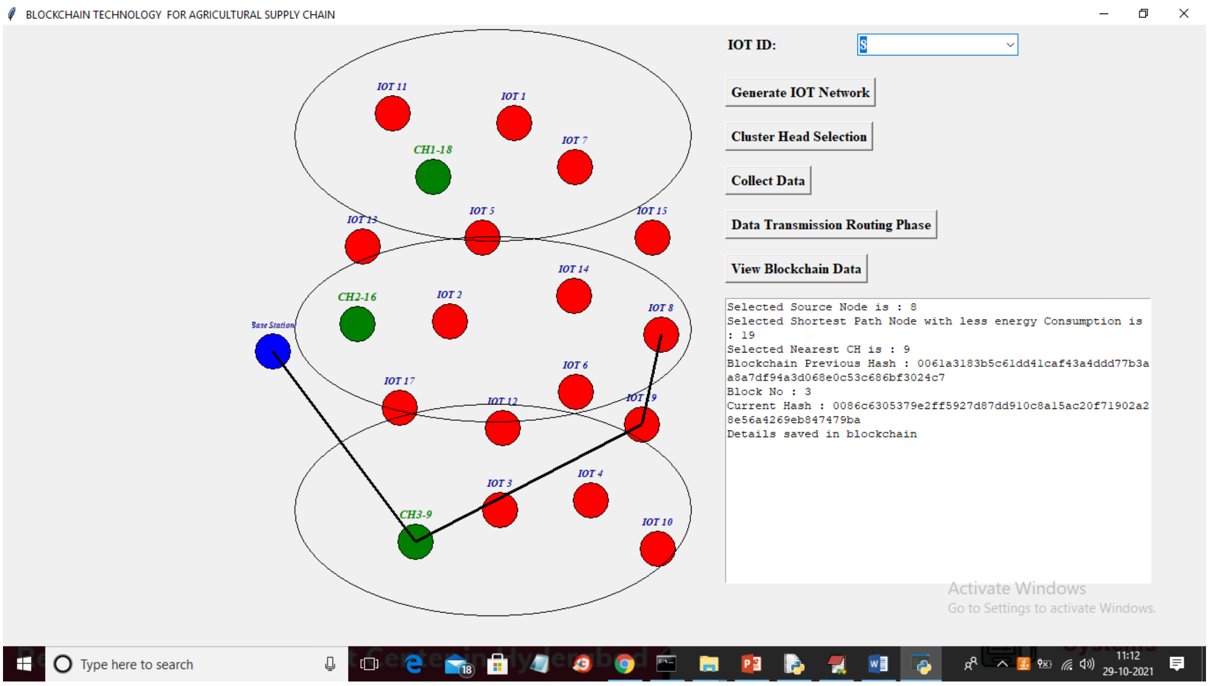


Figure 5.1.6 Data Transmission Routing Phase

In above screen IOT 8 selected nearest hop as 9 and that 9 belongs to CH3 so it send data to CH3 and CH3 is sending data to base station and similarly you can select any IOT and then transfer data.

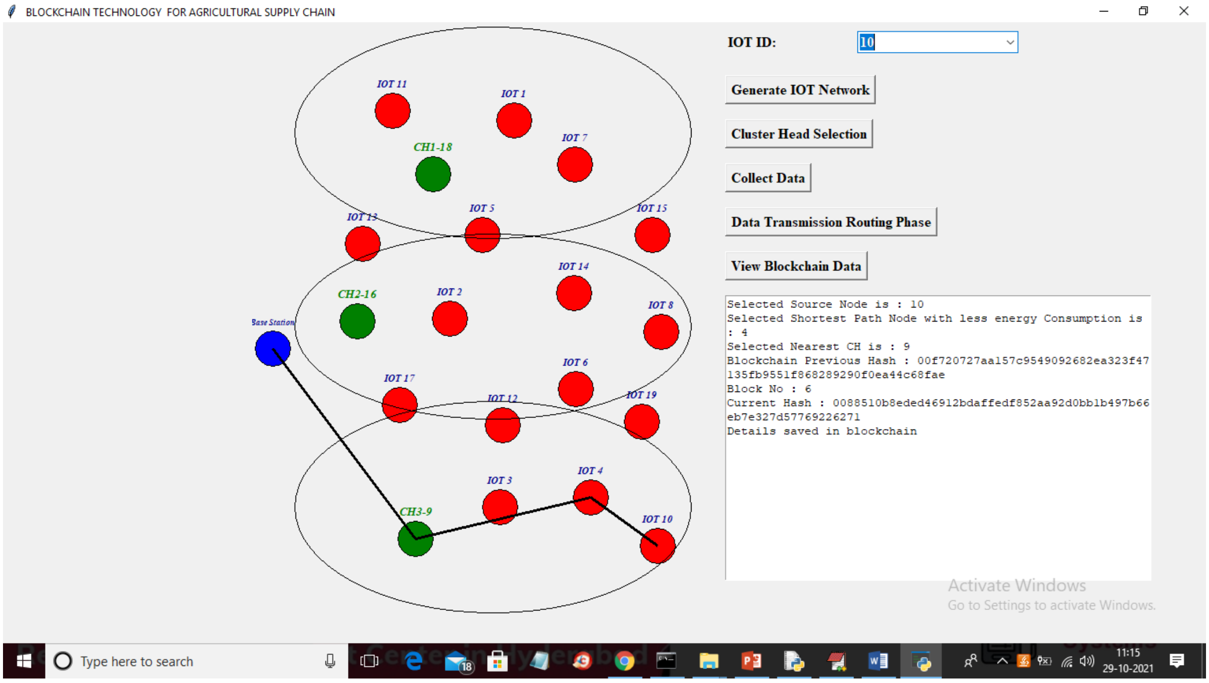


Figure 5.1.7 Transferring Data

In above screen I selected IOT 10 and then it select CH 3 to send data to base station and in above screen we can see each data is stored at Blockchain and each block associated hash code is also displaying and now select any IOT and click on 'View Blockchain Data' to extract data from Blockchain for selected ID

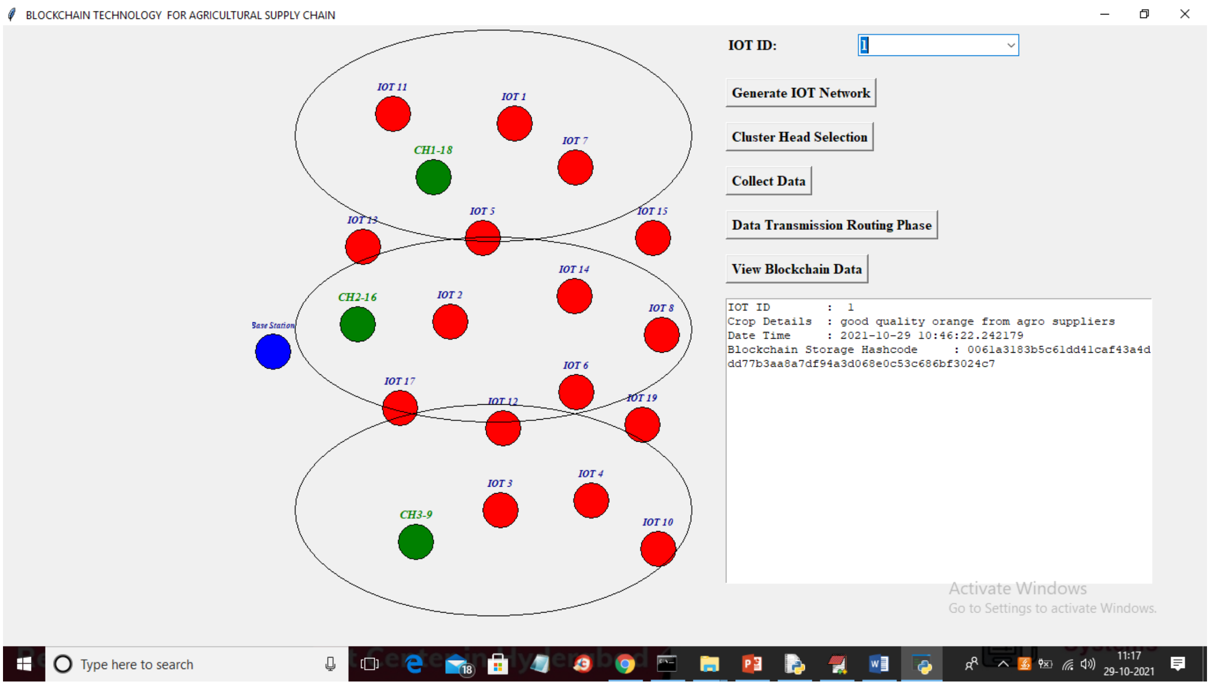


Figure 5.1.8 View Blockchain Data

In above screen I selected IOT as 1 and then clicked on ‘View Blockchain Data’ button and then in text area all data for that IOT retrieve from Blockchain and then displaying and I am displaying hash code of that block.

6.DISCUSSION OF RESULT

The application of blockchain technology in the agriculture supply chain has yielded significant results, promising transformative impacts across various facets of the industry. By leveraging blockchain's capabilities, stakeholders have achieved unprecedented levels of transparency, enabling seamless tracking and authentication of agricultural products from farm to table. This transparency not only mitigates fraud and counterfeiting but also ensures compliance with quality standards and regulations. Moreover, blockchain's immutable ledger facilitates precise traceability, enhancing food safety measures by swiftly identifying and containing any contamination or foodborne illness outbreaks. The efficiency gains brought about by blockchain are equally noteworthy, as streamlined record-keeping and automated processes minimize paperwork, reduce delays, and optimize supply chain operations. Smart contracts, powered by blockchain, automate transactions and enforce agreements, fostering fair trade practices and empowering smallholder farmers by certifying the authenticity and sustainability of their products. Additionally, blockchain analytics offer valuable insights into supply chain performance, enabling stakeholders to identify and address inefficiencies promptly. While these results are promising, challenges such as scalability and interoperability remain, underscoring the need for continued innovation and collaboration to fully harness blockchain's potential in revolutionizing the agriculture supply chain.

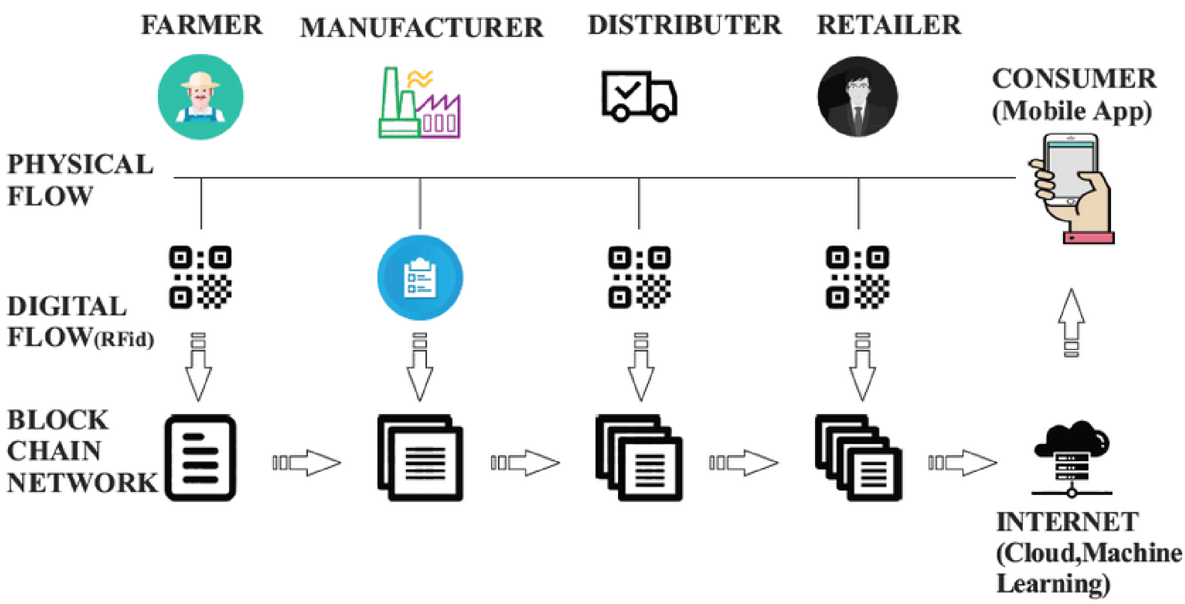


Figure 6.1 Blockchain Network

Blockchain technology has revolutionized traceability in the agriculture supply chain, offering unparalleled transparency and accountability throughout the entire journey of agricultural products. By utilizing blockchain's decentralized ledger, stakeholders can record every transaction and movement of goods in a secure and immutable manner. This level of traceability enables precise tracking of products from their origin on the farm through every stage of processing, distribution, and retail. Each step in the supply chain is documented, providing a comprehensive and verifiable record of the product's journey. This transparency not only enhances food safety by quickly identifying and containing any contamination or quality issues but also instills confidence in consumers by providing them with clear information about the products they purchase. Additionally, blockchain traceability empowers farmers by allowing them to showcase their products' authenticity and adherence to sustainability practices, thereby creating opportunities for market differentiation and premium pricing. Overall, blockchain's traceability capabilities represent a significant advancement in ensuring the integrity and safety of agricultural products while fostering trust and transparency across the supply chain

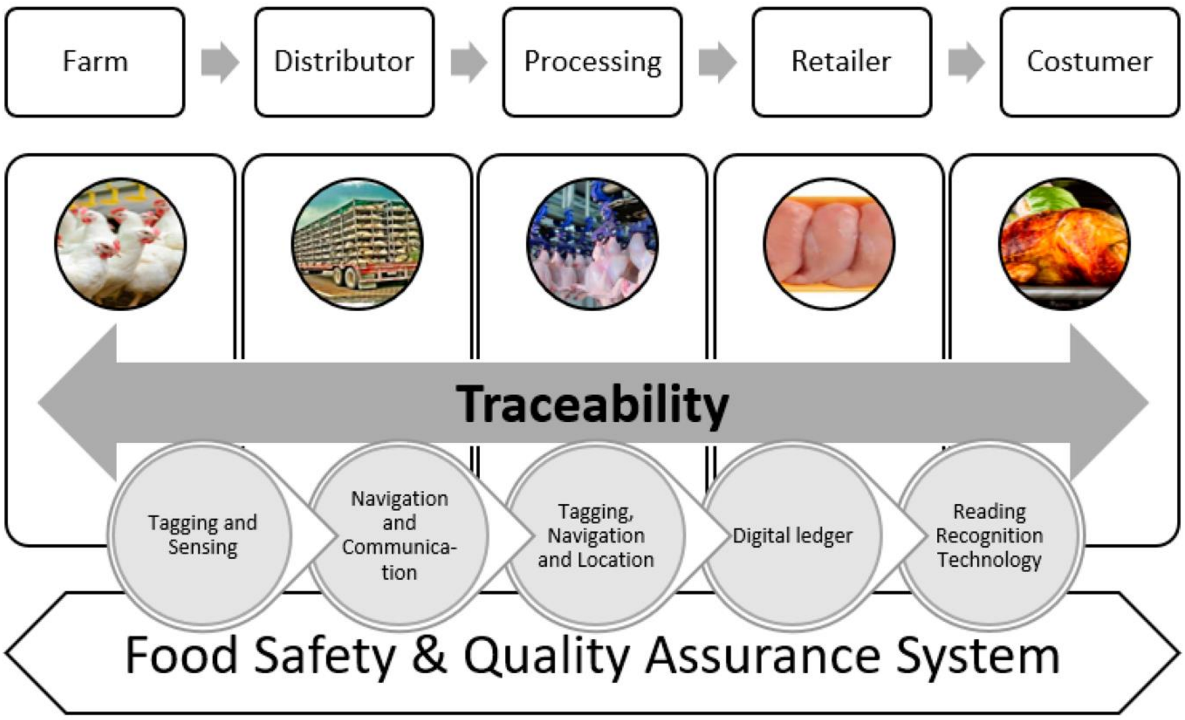


Figure 6.2 Traceability

Blockchain technology has ushered in a new era of transparency within the agriculture supply chain, offering stakeholders unprecedented visibility and accountability at every stage of production, distribution, and consumption. Through the decentralized and immutable nature of blockchain ledgers, information regarding the sourcing, production methods, transportation, and handling of agricultural products becomes readily accessible and verifiable. This transparency enables farmers, processors, distributors, retailers, and consumers to trace the journey of agricultural goods from farm to fork with unparalleled clarity.

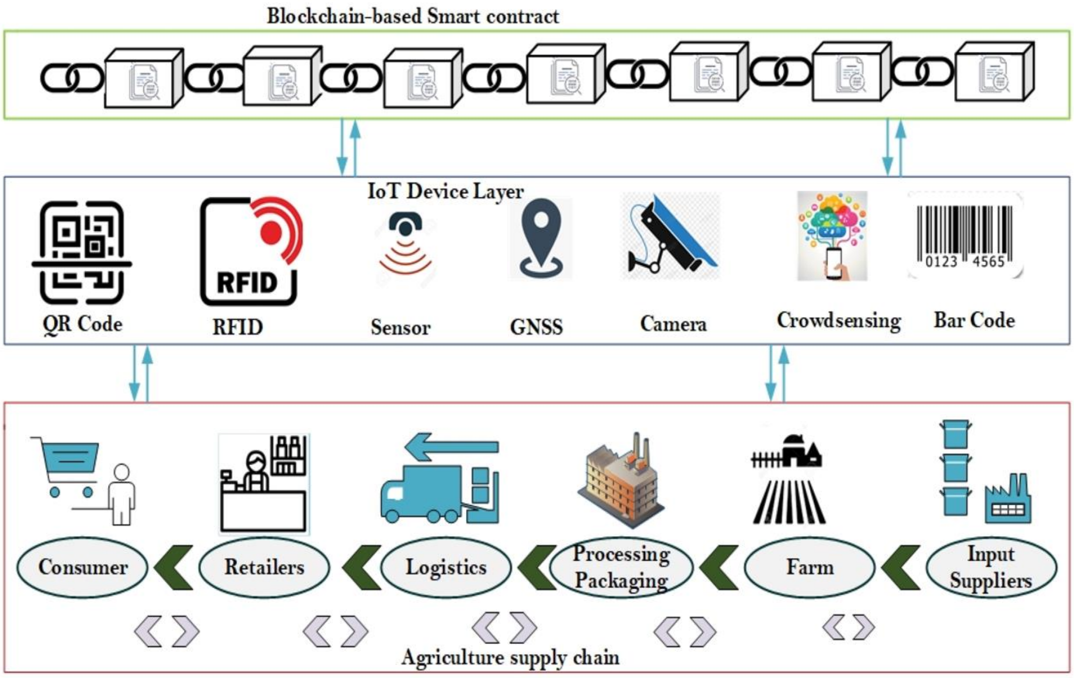


Figure 6.3 Transparency

At the farm level, blockchain allows farmers to digitally record essential data such as planting schedules, irrigation methods, pesticide usage, and harvest yields. This information can then be securely stored on the blockchain, providing transparent insights into farming practices and ensuring compliance with organic or sustainable standards. As the products move through the supply chain, each transaction is recorded on the blockchain, creating an indelible record of ownership, location, and condition.Consumers benefit from this transparency by gaining access to detailed information about the provenance and quality of the products they purchase. With blockchain, they can verify claims such as organic certification, fair trade practices, or adherence to specific dietary preferences or allergens. This level of transparency not only promotes consumer trust but also encourages ethical and sustainable purchasing decision.

7.CONCLUSION

Above Blockchain technology helps in detecting attack nodes and make data secured. In propose work we are using IOT networks and this IOT network implemented following operations successfully, Generate Network: Cluster Head Selection: Collect Data: Data Transmission Routing Phase: View Blockchain Data: In this project we have used IOT sensors and agriculture field but we don’t have any sensors so we built this concept as simulation and analyzed successfully.

Blockchain technology presents a transformative opportunity to reshape the agricultural supply chain. By ensuring transparency, traceability, and trust throughout the food system, blockchain empowers consumers, strengthens relationships between stakeholders, and paves the way for a more sustainable and efficient agricultural future. This technological revolution has the potential to benefit all participants in the food system, from farmers receiving fair compensation to consumers making informed purchasing decisions. As blockchain adoption continues to grow, we can expect a future where our food journey is not just on our plates, but transparently visible from farm to fork.

Blockchain technology has emerged as a revolutionary solution for transforming the agriculture supply chain, offering unprecedented transparency, traceability, and efficiency. By leveraging blockchain, stakeholders across the agricultural sector can seamlessly track the journey of products from farm to fork, mitigating risks, reducing fraud, and ensuring the integrity of the supply chain. Through immutable and decentralized ledgers, farmers, distributors, retailers, and consumers gain access to real-time data on product provenance, quality, and handling practices. This enhanced visibility not only fosters trust and accountability but also enables swift response to potential issues such as contamination or spoilage, thereby safeguarding food safety and public health. Moreover, blockchain facilitates seamless collaboration among supply chain participants, streamlining processes, reducing costs, and unlocking new opportunities for innovation and value creation. As the agricultural industry continues to grapple with challenges such as globalization, climate change, and evolving consumer preferences, blockchain stands as a beacon of hope, offering a robust foundation for building resilient, sustainable, and equitable food systems of the future. In conclusion, the integration of blockchain technology into the agriculture supply chain holds immense promise in revolutionizing the way we produce, distribute, and consume food, ushering in a new era of transparency, trust, and efficiency.

8.REFERENCE

[1] Aiken A. Zooming in on privacy concerns: Video app Zoom is surging in popularity. In our rush to stay connected, we need to make security checks and not reveal more than we think. Index Censorsh. 2020;49(2):24–27.

[2] Bermeo-Almeida O., Cardenas-Rodriguez M., Samaniego-Cobo T., Ferruzola-Gómez E., Cabezas-Cabezas R., Bazán-Vera W. International Conference on Technologies and Innovation, 6-9 November 2018. Guayaquil; Ecuador: 2018. Blockchain in agriculture: a systematic literature review; pp. 44–56.

[3] Brewin D. The impact of COVID-19 on the grains and oilseeds sector. Can. J. Agric. Econ./Rev. Can. Agroecon. 2020;68:185–188.

[4] Chofreh A.G., Goni F.A., Jofreh M.G. Enterprise resource planning (ERP) implementation process: project management perspective. Adv. Mater. Res. 2011;338:152–155.

[5] Chofreh A.G., Goni F.A., Shaharoun A.M., Ismail S. A review on sustainability transformation roadmaps using project management methodology. Adv. Sci. Lett. 2015;21(2):133–136.

[6] Chofreh A.G., Goni F.A., Klemeš J.J. A master plan for the implementation of sustainable enterprise resource planning systems (Part II): development of a roadmap. Chem. Eng. Trans. 2016;52:1099–1104.

[7] Chofreh A.G., Goni F.A., Malik M.N., Khan H.H., Klemeš J.J. The imperative and research directions of sustainable project management. J. Clean. Prod. 2019;238A

[8] Chofreh A.G., Goni F.A., Klemeš J.J., Moosavi S.M.S., Davoudi M., Zeinalnezhad M. Covid-19 shock: development of strategic management framework for global energy. Renew. Sustain. Energy Rev. 2020;139

[9] CTA . 2017. Perspectives for ICT and Agribusiness in ACP countries: Start-up financing, 3D printing and blockchain.https://www.cta.int/en/event/perspectives-for-ict-and-agribusiness-in-acp-countries-start-up-financing-3d-printing-and-blockchain-sid002d57e47-75f4-4837-af9b-8fd5d10d5162>

[10] de Paulo Farias D., dos Santos Gomes M.G. COVID-19 outbreak: what should be done to avoid food shortages? Trends Food Sci. Technol. 2020;102:291–292.

9.APPENDIX

PYTHON

9.1 Introduction

One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

- **Readability:** Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
- **Easy to Learn:** Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
- **Interpreted Language:** Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
- **Large Standard Library:** Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O,

networking, regular expressions, databases, and more, saving developers time and effort.

- **Object-Oriented:** Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.

9.2 Installation

To install Python on your computer, follow these basic steps:

- **Step 1:** Visit the Python website Go to the official Python website at https://www.python.org/.
- **Step 2:** Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
- **Step 3:** Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
- **Step 4:** Install Python 3.7.0 using Cmd.
- **Step 5:** Install the all libraries that required to run the project
- **Step 6:** Run

9.3 Python GUI (Tkinter)

Python provides a wide range of options for GUI development (Graphical User Interfaces).Tkinter, the most widely used GUI technique, is used for all of them. The Tk GUI toolkit offered by Python is used with the conventional Python interface. Tkinter is the easiest and quickest way to write Python GUI programs. Using Tkinter, creating a GUI is simple.
A part of Python's built-in library is Tkinter. The GUI programs were created. Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter. Making a GUI application is easy using Tkinter.
Following are the steps:

- **Step 1:** Install the Tkinter module in place.
- **Step 2:** The GUI application Makeske the primary window
- **Step 3:** Include one or more of the widgets mentioned above in the GUI application.
- **Step 4:** Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

9.4 Python IDLE

Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.

IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.

IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

- Open the command prompt (Windows) or terminal (macOS/Linux).
- Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
- Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

9.5 Libraries

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

There are some libraries following:

9.5.1 Pandas:

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis. Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

- It has functions for analysing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
- Pandas allow us to analyse big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.

9.5.2 NumPy:

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

9.5.3 Matplotlib:

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

9.5.4 Tensor flow

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

9.5.5 Tkinter

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

- Cross-Platform Compatibility
- Simple and Easy-to-Use
- Widgets and Layout Management
- Event-Driven Programming
- Customization and Styling
- Integration with Other Libraries