# PROJECT REPORT

# NAAN MUDHALVAN

**PROJECT NAME: Electronic Voting System**

**COURSE NAME:** Block Chain Technology

This Project is done by:


*Team Leader Name:* B.Lakshminarayana

*Team Member 01 :* Chandhu.E

*Team Member 02 :* Gowtham.P.S

*Team Member 03 :* Harsha Vardhan

*Department:* Electronics and Communication Engineering *College Name & Address:*

Prathyusha Engineering College
Poonamallee-Tiruvallur Road, Aranvoyal Kuppam,

Tiruvallur Taluk

PINCODE: 602025


## 1. INTRODUCTION

An electronic voting system using blockchain technology is a revolutionary approach to conducting elections in a secure, transparent, and efficient manner. This system leverages the unique properties of blockchain technology to address the challenges faced by traditional voting methods

## 1.1. Project Overview

**Blockchain Technology**: Blockchain is a type of distributed ledger technology (DLT) based on computer science, cryptography, and mathematics[2]. It operates across a global network of computers called "nodes", which are responsible for verifying, managing, and storing data or financial transactions

**Challenges with Traditional Voting**: Traditional voting systems like ballot box voting or electronic voting suffer from various security threats such as DDoS attacks, polling booth capturing, vote alteration and manipulation, malware attacks, etc.They also require huge amounts of paperwork, human resources, and time[1]. These issues create a sense of distrust among existing systems

**Advantages of Blockchain Voting**: By leveraging blockchain technology, the voting process can be made more secure, transparent, immutable, and reliable Blockchain's immutability ensures that once a vote is cast, it cannot be altered or removed. Its transparency allows voters to verify that their votes have been counted correctly[1]. Moreover, blockchain-based e-voting eliminates the need for physical verification of voters at polling stations, reducing the potential for malpractice

**Decentralization**: Building a blockchain system is not enough. It should be decentralized i.e if one server goes down or something happens on a particular node, other nodes can function normally and do not have to wait for victim node's recovery

In conclusion, an electronic voting system using blockchain technology has the potential to significantly improve the overall security and efficiency of elections. However, as blockchain is still a relatively niche and developing technology, there are some technical hurdles to overcome before we see a blockchain voting system in mainstream electoral polls[2].

### 1.2. Purpose

The purpose of this project is to design and implement a secure, transparent, and efficient electronic voting system using blockchain technology. This system aims to overcome the challenges faced by traditional voting methods, enhancing the overall integrity of elections.

# 2.IDEATION & PROPOSED SOLUTION

### 2.1.Problem Statement Definition

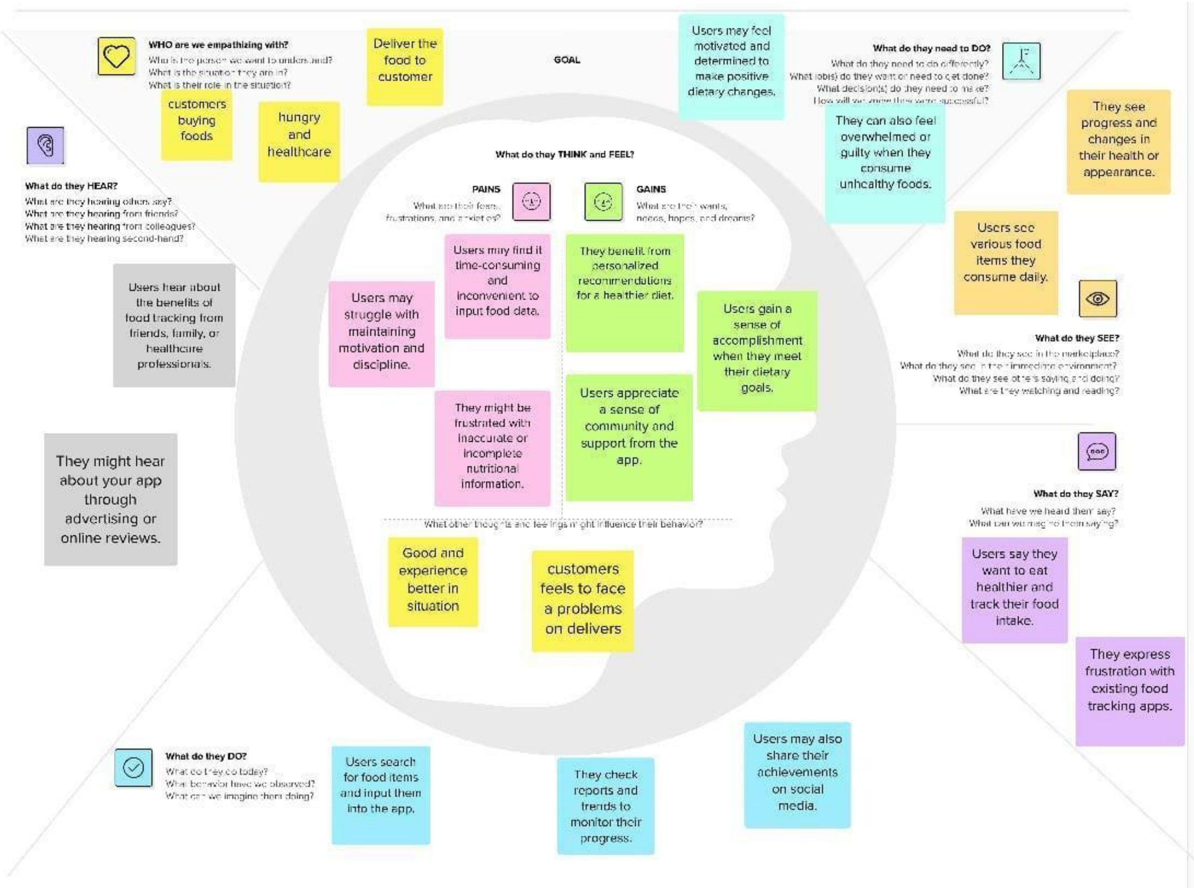
The problem statement for an electronic voting system using blockchain technology can be defined as follows:

"Current voting systems, both physical and digital, are plagued with numerous issues such as security threats, lack of transparency, potential for manipulation, and inefficiency. These challenges undermine the integrity of elections and create a sense of distrust among voters[1]. The need of the hour is a secure, transparent, and efficient voting system that can safeguard the democratic rights of the people. This project aims to address these issues by leveraging blockchain technology to develop a decentralized electronic voting system."

### 2.2.Empathy Map Canvas

1. **Users (Voters)**:
   - **Think & Feel**: Voters may feel skeptical about the security and transparency of traditional voting systems[12]. They might be excited about the potential of blockchain technology to improve these aspects[12].
   - **See**: Voters see news about election fraud, vote manipulation, and lack of transparency in traditional voting systems[12].
   - **Say & Do**: Voters express their concerns about traditional voting systems and show interest in exploring new technologies like blockchain for voting[12].
   - **Hear**: Voters hear about the potential of blockchain technology to revolutionize various sectors, including voting[12].

2. **Election Authorities**:
   - **Think & Feel**: Election authorities might be concerned about the security threats and inefficiencies associated with traditional voting methods[12]. They could be optimistic about the potential of blockchain technology to address these issues[12].
   - **See**: Election authorities see instances of election fraud, vote manipulation, and inefficiencies in traditional voting systems[12].
   - **Say & Do**: Election authorities express the need for more secure, transparent, and efficient voting systems. They might be open to exploring and implementing blockchain technology for voting[12].
   - **Hear**: Election authorities hear about successful implementations of blockchain technology in various sectors and its potential benefits for voting[12].
3. **Developers**:
   - **Think & Feel**: Developers might be excited about the technical challenges and opportunities associated with developing a blockchain-based electronic voting system[12].
   - **See**: Developers see the growing interest in blockchain technology and its potential applications in various sectors, including voting[12].
   - **Say & Do**: Developers express their interest in blockchain technology and work on developing secure, transparent, and efficient blockchain-based electronic voting systems[12].
   - **Hear**: Developers hear about the latest advancements in blockchain technology and its potential to revolutionize various sectors, including voting[12].




## 2.3.Ideation & Brainstorming

Ideation and brainstorming for an electronic voting system using blockchain technology could involve the following points:

1. **Understanding the Problem**: The first step is to understand the challenges with the current voting systems, such as security threats, lack of transparency, potential for manipulation, and inefficiency[12].
2. **Exploring Blockchain Technology**: Next, explore how blockchain technology can address these issues. Blockchain's properties like security, transparency, immutability, and decentralization make it a promising solution[12].
3. **Designing the System**: Brainstorm on how to design the system. This could involve using smart contracts to develop a decentralized voting application (DApp) on the Ethereum blockchain[2]. The implementation might consist of a web interface for users to interact with the blockchain[12].
4. **Addressing Challenges**: Discuss potential challenges and how to overcome them. These could include technical hurdles related to the

scalability and performance of the blockchain, as well as legal and regulatory issues[2].

5. **Testing and Iteration**: Finally, brainstorm on how to test the system and iterate based on feedback. This could involve conducting pilot tests during smaller elections before scaling up[12].

Remember, ideation and brainstorming is a creative process that involves open discussion, collaboration, and iteration.

### 2.4. Proposed Solution

1.Proposed below is a high-level reference architecture for Online Food Delivery systems.

2.This proposed architecture is generic and it can be deployed to any of cloud provider like AWS/GCP/Azure

Assumptions:

Delivery partners are provided with devices that have inbuilt GPS. This will help to locate their current location accurately. Integration with map provider is there and we get details about routes, traffic and commute time.

## 3.REQUIREMENT ANALYSIS

### 3.1.Functional Requirements

1. User Registration and Profile Management:

Users should be able to create accounts, set up profiles, and manage their personal information.

2. Food Logging:

Users can log their daily food intake, including details like quantity, preparation method, and mealtime.

3. Food Database:

The system must have a comprehensive database of foods, ingredients, and recipes, allowing users to easily select and log items.

4. Nutritional Analysis:

The system should calculate and display nutritional information, such as calories, macronutrients (carbohydrates, fats, proteins), and micronutrients (vitamins and minerals) for each food item and overall meals.

5. Barcode Scanner:

Implement a feature for users to scan barcodes on packaged food items for quick input of nutritional data.

6. Meal Planning:

Users should be able to plan meals in advance, including breakfast, lunch, dinner, and snacks, with the ability to set portion sizes and nutritional targets.

7. Personalized Goals:

Allow users to set personalized nutritional goals based on factors like age, weight, gender, and activity level.

8. Recommendations and Insights:

Provide personalized recommendations and insights to help users make healthier food choices based on their goals and dietary restrictions.

## 3.2.Non-Functional Requirements

1. Data Privacy and Security:

Ensure that user data, including personal information and dietary records, is securely stored and protected from unauthorized access.

2. Scalability:

Design the system to handle a growing user base and increasing data volumes without performance degradation.

3. Usability and User Experience:

The system should have an intuitive and user-friendly interface to encourage regular use by a diverse user base.

4. Reliability:

The system must be available and operational 24/7, minimizing downtime for maintenance.

5. Performance:

Ensure quick response times for data retrieval, calculations, and other core functions, even during peak usage.

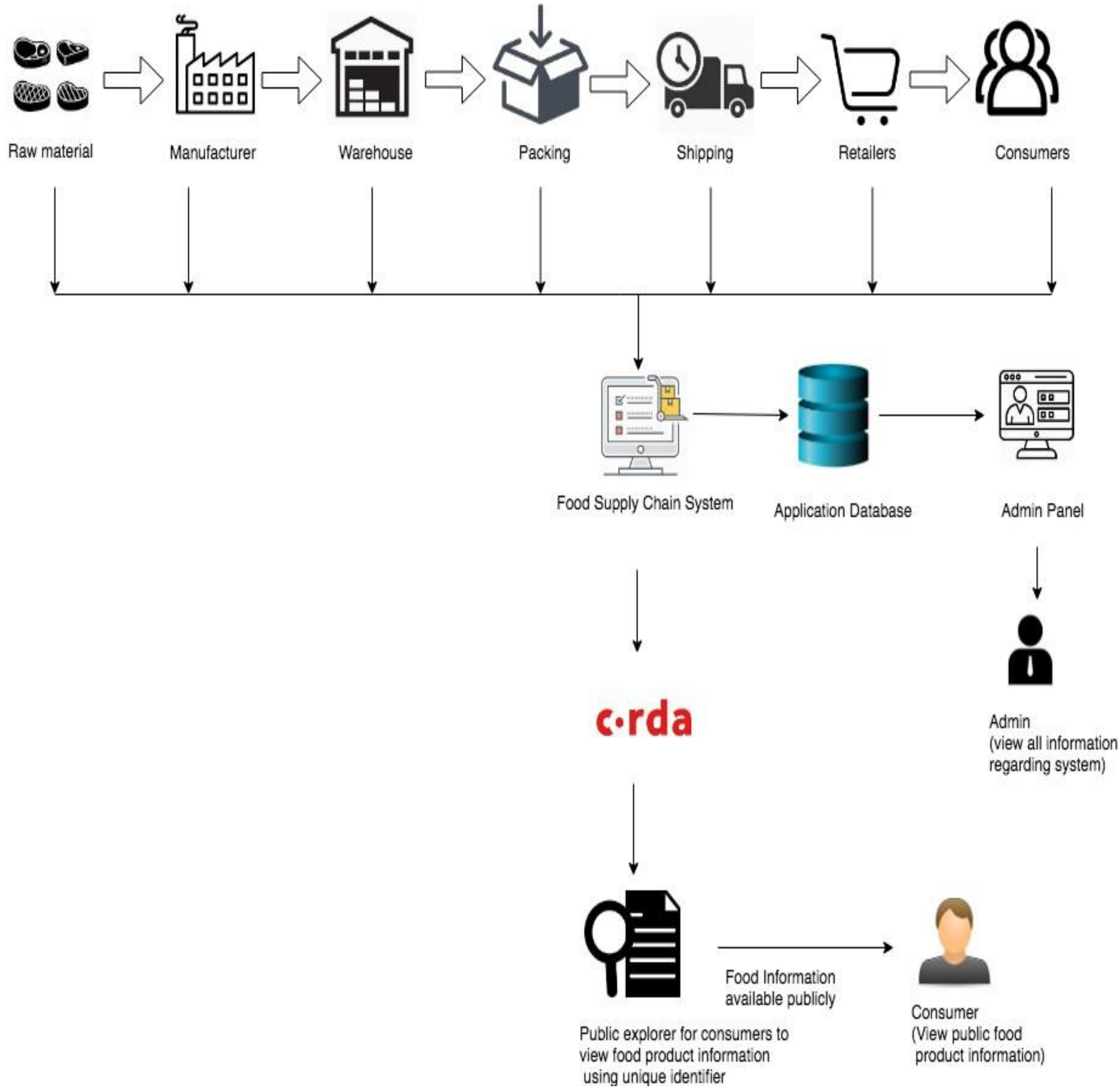
6. Compatibility:

The system should be compatible with various devices and platforms, such as web browsers, mobile devices, and wearable tech.

# 4.PROJECT DESIGN

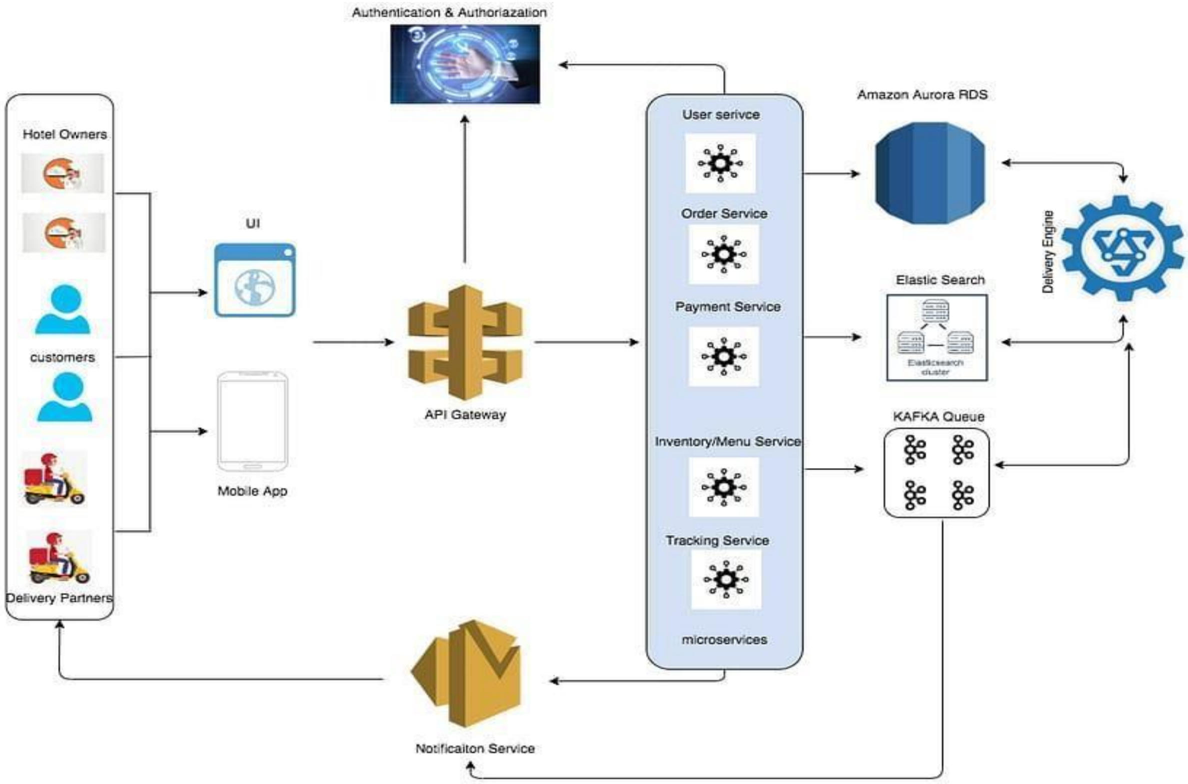
## 4.1.Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

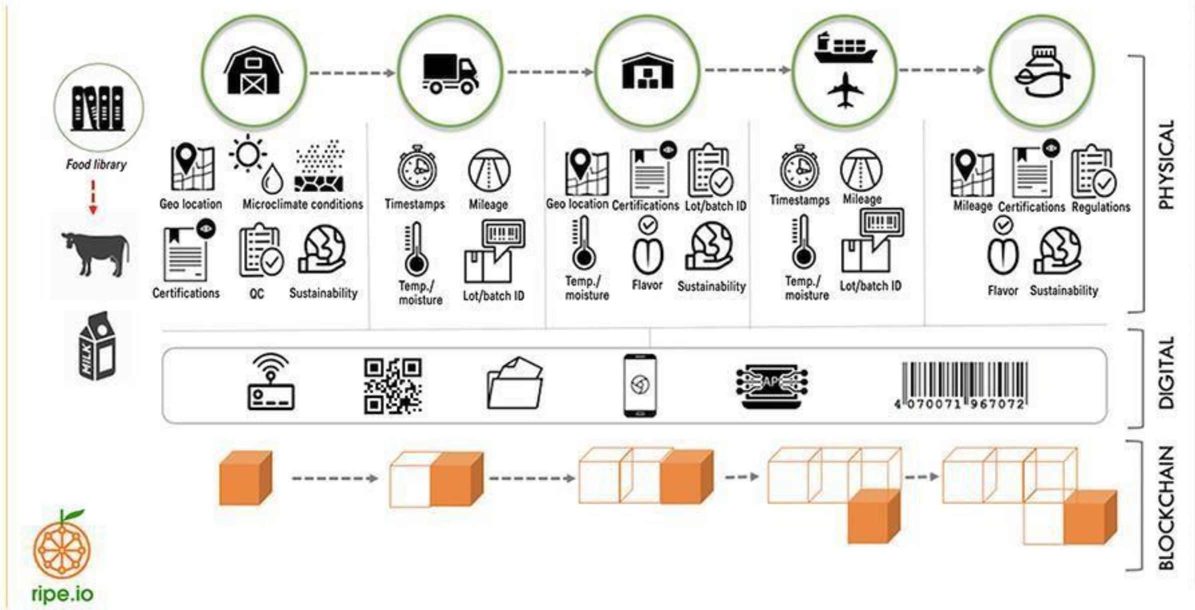



## 4.2.Solution & Technical Architecture

Solution architecture is a complex process – with many subprocesses – that bridges the gap between business problems and technology solutions.




## 4.3.User Stories

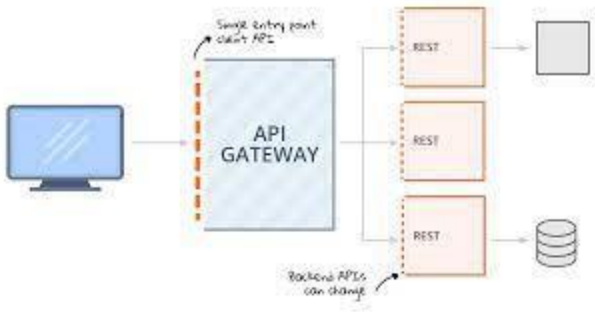



# 5.CODING & SOLUTION

## 5.1.Feature 1 API

Gateway:

An API gateway is a data-plane entry point for API calls that represent client requests to target applications and services. It typically performs request processing based on defined policies, including authentication, authorization, access control, SSL/TLS offloading, routing, and load balancing.




## 5.2.Feature 2

### Authentication,Authorization & Payment:

Authentication is knowing the identity of the user. For example, Alice logs in with her username and password, and the server uses the password to authenticate Alice. Authorization is deciding whether a user is allowed to perform an action. For example, Alice has permission to get a resource but not create a resource.

## Code:

```
{
  "name": "food-tracking",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
```

```
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^5.3.1",
    "ethers": "^5.6.6",
    "react": "^18.2.0",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

## 5.3.Data Base Scheme

Here we used Block Chain Technology for storing the data for more security for the data.

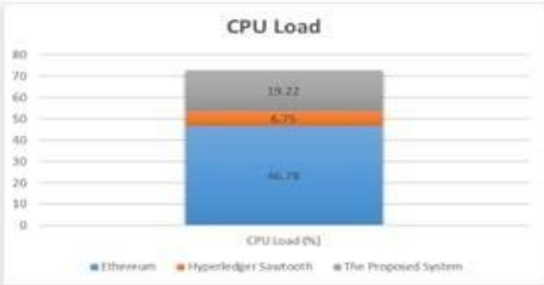
# 6.RESULTS

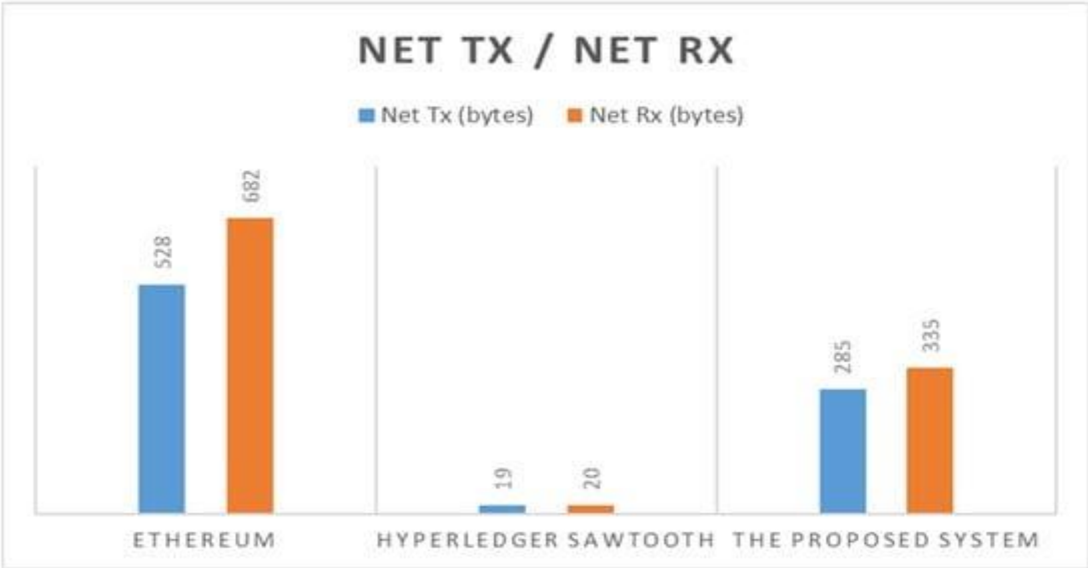
Before using the blockchain-based food tracking system, the performance data of the system were obtained. In this way, it will be necessary to prevent problems such as scalability and to stop the work if it is foreseen that the blockchain-based system to be used will not reach the desired performance values. The performance values of Ethereum and Hyperledger Sawtooth are used to benchmark the values obtained from the proposed system. A simulation environment has been set up to collect and compare these data using Matlab. The latency (s), Net Tx (bytes), Net Rx (bytes), and CPU load (%) values are the variables that keep the data obtained in this simulation environment. With the data obtained in this simulation environment, the aim is to reveal the difference with other platforms clearly and concretely.

The latency (s) value in the proposed system was obtained as 0.038. The transmission per second value is 285, the reception per second value is 335, and the CPU load rate value is 19.22. Especially when we evaluate the latency times, the obtained value is at a very good level compared to Ethereum. When it is compared with Hyperledger Sawtooth, it is seen that there is a little more delay. The main reason for this is that the system architecture is more complicated, and the data size obtained is high. This is also evident from the fact that the transmission per second and reception per second values are much higher than Hyperledger Sawtooth. It has been observed that a rate of 19.22 was achieved in the CPU usage rate As a result, it is seen that the performance data obtained have a serious advantage over Ethereum, especially in terms of latency, and it has started to converge in other values. Considering that the realtime operation of the installed system is extremely important, the choice of Hyperledger Fabric has once again emerged as the right decision.

## 6.1.Performance Metrics

Graphical Representation:


# 7.ADVANTAGES & DISADVANTAGES

Advantages:

- Improved Food Safety
- Rapid Recall
- Quality Control
- Transparency
- Regulatory Compliance

Disadvantages:

- Implementation Costs
- Data Management
- Resistance to Change

- Complexity
- Error Potential

# 8.CONCLUSION

In conclusion, while food traceability systems offer numerous advantages in terms of food safety, quality, transparency, and efficiency, they also come with challenges related to cost, complexity, and data management. To maximize the benefits and overcome the disadvantages, businesses must carefully plan and implement traceability systems that align with their specific needs and goals.

# 9.FUTURE SCOPE

The future scope for food traceability systems is promising, with several trends and developments driving their continued growth and evolution. Here are some key areas of future scope for food traceability systems:

1. **Blockchain Technology**: Blockchain offers a secure, transparent, and decentralized way to record and verify food supply chain data. It can provide an immutable ledger of every step in the supply chain, enhancing transparency and trust. Many companies are exploring blockchain-based traceability solutions.
2. **Internet of Things (IoT)**: IoT devices can be used to track and monitor food products throughout the supply chain. Sensors can collect data on temperature, humidity, and other environmental conditions to ensure food quality and safety.
   This real-time data can be integrated into traceability systems.
3. **Artificial Intelligence (AI)**: AI can analyze large datasets generated by traceability systems to identify patterns and anomalies. Machine learning algorithms can help predict and prevent food safety issues and optimize supply chain operations.
4. **Smart Packaging**: Packaging with embedded RFID (RadioFrequency Identification) tags or QR codes can provide consumers with information about the product's source, expiration date, and handling instructions. Smart packaging can also enable easier tracking throughout the supply chain.

5. **Consumer Demand for Transparency**: Increasing consumer demand for transparency and information about the food they consume will drive the adoption of more comprehensive and user-friendly traceability systems. Consumers may expect to access detailed information via mobile apps or websites.
6. **Regulatory Requirements**: Governments and regulatory bodies are likely to continue strengthening food safety regulations, including traceability requirements. This will drive the adoption of more robust and standardized traceability systems.
7. **Globalization and Supply Chain Complexity**: As supply chains become more global and complex, the need for traceability systems to manage and secure these supply chains will increase. This includes traceability of both raw materials and finished products.
8. **Sustainability and Ethical Sourcing**: Traceability systems can help track and verify sustainable and ethical sourcing practices, addressing growing consumer concerns about environmental and social responsibility in the food industry.
9. **Third-Party Verification**: Independent third-party organizations specializing in food traceability verification may play a larger role in certifying the accuracy and reliability of traceability data, further enhancing trust in the system.
10. **Integration with Emerging Technologies**: Traceability systems will likely integrate with emerging technologies like 5G, edge computing, and augmented reality, enabling faster data transfer and real-time insights throughout the supply chain.
11. **Food Waste Reduction**: Traceability systems can be used to reduce food waste by providing data to optimize inventory management, reduce spoilage, and ensure products are used before their expiration dates.
12. **Customization and Scalability**: Future traceability systems will need to be customizable and scalable to fit the specific needs of different sectors within the food industry, from small-scale farms to large food conglomerates.

# 10.APPENDIX

1. **Sample Reports** :
   - Mock food consumption reports generated by the system to demonstrate its capabilities.
2. **User Guides** :
   - User manuals or guides explaining how to use the food tracking system.
   - Troubleshooting guides for common issues.
3. **Data Tables** :
   - Tables of food items with nutritional information.
   - Sample data sets for testing and understanding data input and output.
4. **Database Schema** :
   - An overview of the database structure used in the system, including tables, fields, and relationships.
5. **Flowcharts**:
   - Visual representations of the data flow and user interactions within the system.
6. **Sample Input Screens**:
   - Screenshots or wireframes illustrating the data entry forms or interfaces for users.
7. **Terms and Definitions**:
   - A glossary of key terms related to the food tracking system, such as "calories," "macros," "serving size," etc.
8. **Privacy and Security Policies**:
   - Information on how user data is collected, stored, and protected in compliance with relevant laws and regulations.
9. **API Documentation**:
   - If the system offers an API for developers, include documentation on endpoints, request/response formats, and authentication methods.
10. **Feedback Forms**:
    - Templates for collecting user feedback and suggestions for improvement.

11. **Legal Disclaimers**:
    - Legal information regarding terms of use, copyright, and liability.
12. **System Architecture**:
    - Diagrams or explanations of the system's architecture and components.
13. **Scalability and Performance Information**:
    - Details on how the system handles increased loads and performance benchmarks.
14. **Support Contact Information**:
    - Contact details for user support, including email addresses, phone numbers, or a support ticket system.
15. **Release Notes**:
    - A history of system updates and changes, including new features, bug fixes, and improvements.
16. **References**:
    - Citations and references to academic studies, nutritional databases, or other sources used in developing the system.

# GitHub Project Link: