K S R INSTITUTE FOR ENGINEERING AND TECHNOLOGY

TIRUCHENGODE: 637 215

Computer Science and Engineering

NAAN MUDHALVAN

SB8024- Blockchain Development

by Naan Mudhalvan Scheme – 2023

TEAM ID: NM2023TMID

PROJECT DOMAIN: BLOCKCHAIN TECHNOLOGY

PROJECT TITLE: FOOD TRACKING SYSTEM

TEAM MEMBERS

REGISTER NUMBER	NAME
731620104002	AJITH S
731620104004	ARUL MURUGAN V
731620104010	DEVARAJ M
731620104047	SANJAY N

TABLE OF CONTENT

1.INTRODUCTION

In the landscape of modern education, the management of student credentials stands at a crossroads. The traditional methods of record-keeping, often paper-based and centralized, are proving to be increasingly inadequate in the digital age. With the advent of blockchain technology, a groundbreaking solution has emerged: Transparent Education Data Management. At its core, this innovative approach aims to reimagine how academic certificates and student records are stored, accessed, and verified. By harnessing the decentralized power of blockchain, a distributed ledger technology known for its immutable and transparent nature, the education sector can embark on a transformative journey.

In this paradigm, digital certificates of students find their home within a distributed and decentralized network, offering a level of security and integrity previously unseen. Blockchain's foundational characteristic of immutability ensures that once data is entered, it remains unalterable, thereby eliminating the risks associated with tampering and forgery. The process is made even more efficient through the incorporation of smart contracts, self-executing agreements with pre-defined rules. These contracts not only enable the seamless addition of certificate details to the blockchain but also ensure the automation of verification processes, reducing the bureaucratic hurdles faced by institutions and employers.

Crucially, Transparent Education Data Management extends beyond mere storage; it emphasizes the concept of real-time verification. Through the integration of decentralized storage networks like IPFS, the actual certificate files are securely housed. This dual-layered approach guarantees the integrity of the certificates while allowing for swift and accurate verification. Employers, academic institutions, and other stakeholders can query the blockchain, instantaneously confirming the authenticity of a student's credentials.

1.1 Project Overview

The Transparent Education Data Management on blockchain project aims to provide a robust and secure solution for storing and managing digital certificates of students within a distributed and decentralized network. This innovative system leverages blockchain technology to ensure the integrity, accessibility, and transparency of educational certificates.Traditional paper-based certificates and digital records often suffer from issues like fraud, tampering, and loss. This project seeks to address these challenges by implementing a blockchain-based infrastructure that offers immutable, secure, and easily verifiable storage and retrieval of certificate details.

The system will allow educational institutions, certification authorities, and other relevant entities to seamlessly add certificate details to the blockchain. This includes information such as the student's name, institution, degree earned, date of issuance, and any other relevant data. Through the blockchain, this information is recorded in a manner that cannot be altered, ensuring the long-term validity of the certificates.Furthermore, the project will facilitate the easy and efficient querying of certificate details from the blockchain. By leveraging the blockchain's decentralized nature, stakeholders will be able to retrieve certificate information quickly and reliably. This will help in streamlining processes such as background checks, verification of qualifications, and academic credential assessments.

The use of blockchain technology in this context ensures transparency as the blockchain ledger is open for verification by authorized parties. This will help mitigate fraudulent activities associated with counterfeit certificates. Additionally, the decentralized network structure guarantees that the system is resistant to centralized control and single points of failure.

1.2 Purpose

The purpose of the Transparent Education Data Management on blockchain solution is to revolutionize the way educational certificates are handled and managed, addressing several critical challenges in the field of education and certification. This innovative project primarily aims to provide a secure, decentralized, and immutable platform for storing and retrieving digital certificates of students. One of the core objectives of this solution is to enhance the security and integrity of educational certificates. Traditional paper-based certificates or even digital records are susceptible to fraud, tampering, and loss. By leveraging blockchain technology, the system ensures that once certificate details are added to the blockchain, they become immutable and resistant to any unauthorized changes. This enhances the trustworthiness of educational credentials, making it significantly more difficult for individuals to create counterfeit certificates or misrepresent their qualifications.

Furthermore, the project seeks to improve the accessibility and efficiency of managing educational certificates. The decentralized nature of the blockchain allows multiple stakeholders, including educational institutions, employers, and certificate verification agencies, to access and query certificate details with ease. This reduces the administrative burden on educational institutions and simplifies the process of verifying academic qualifications for both employers and students.

Transparency is another crucial aspect of this solution. By utilizing a distributed ledger, the system provides transparency in the management of certificate data. Authorized parties can independently verify the authenticity of certificates without relying on a centralized authority. This transparency is particularly valuable for employers and institutions conducting background checks and for individuals seeking to have their qualifications recognized internationally.

2.LITERATURE SURVEY

“Smart Contract enabled Online Examination System Based in Blockchain Network”, Author : Naresh Chandra,P.Chinnasamy,Year:2021,Publisher:IEEE

Blockchain is the prominent technology that can provide the security and loyalty of data. Initially, blockchain has been used for the cryptocurrency and all the data were available on the public distributed ledger. But now a days, private blockchain is widely used within the organizations for data security. Blockchain engenders decentralized systems in which data can be send and receive securely and efficiently over the network. It means there is everything is hidden from the outer world; only authorized users have the authority to read and write the data on the network. World's topmost industries; like Walmart, IBM, Google, etc. are adopting the blockchain technology to build the Decentralized Applications (DApps). Decentralized Applications are the smart systems that are executed on a distributed computer network. Blockchain enables one of the most secure applications called Smart Contract. Smart Contracts are the computerized and secured distributed ledgers that enable secure, transparent, and tamper-proof transactions. Smart contracts create and verify the data with the help of hashing. It is a mathematical procedure that uses the most powerful algorithm cryptographic Hash Algorithm i.e. SHA-256. It engenders 256-bit signature for the input text. Ethereum Blockchain Platform is a widely used platform to build the DApps. This platform is a public network platform, which is open to all and anyone can participate in this network to send and receives the transactions.We have developed an application for the Online-Examination System using Blockchain Ethereum Platform with features of Smart Contracts that enables server runtime environment NodeJS.

“Decentralized Accreditation of Educational Attainments using Blockchain”,Author:Sumithra V,Shasidhara R,Year:2021,Publisher:IEEE

Today's world is connected through the digital platforms Independent of Geographical location and to serve the needs of the administration process involved in educational we need the platform that help to work through network in a secured and transparent manner and due to the Blockchain core behaviors like transparent, confidentiality, Immutable, decentralized and distributed nature, the Legacy Administrative process for Students Certificate management and verification can be replaced. The Proposed model also helps to setup a streamlined administration process in the educational institute solve the variety of issues that are there in legacy educational organization/institutes administration process like tracking of records in cloud, fraud certification detection, keeping Identity digital under the control of authentic source and transferring over the internet securely, by eliminating need to verify them by third party sources. In this paper we proposed a system to solve the current cumbersome administration process by reducing the huge paperwork involved and by reducing the cost of double spend for manpower among the participating, reduce the wait time for the involved stake holders like Students, guardians and universities. A blockchain-based system, which manages and controls virtual internationalization in higher education institutions, is proposed. The blockchain reporting system has shown promising results in management of international projects with considerable reduction in time and cost. Blockchain technology offers a unique opportunity to manage virtual internationalization activities and actions in a secure, transparent, and immutable fashion. The system consists of administrative and financial components for reporting Erasmus project activities, including staff costs and reimbursement, mobility and travel expenses, equipment, printing and publishing, indirect and other costs.

2.1 Existing Problem

The implementation of Transparent Education Data Management on a blockchain, as described, aims to address several existing problems and challenges in the traditional management of digital certificates and educational records. One of the significant problems in the current education data management is the prevalence of certificate fraud and tampering. Verifying the authenticity of academic credentials is often a time-consuming and cumbersome process. Educational institutions, employers, and verification agencies may face difficulties in confirming the legitimacy of certificates, which can result in delays in recruitment, admissions, or other processes. Many educational institutions maintain centralized databases of student records and certificates. These centralized systems can be vulnerable to data breaches, and they may not provide easy access to graduates who need to share their credentials with multiple parties. A distributed and decentralized approach can alleviate the risks associated with centralization. Protecting the privacy and security of educational records is a paramount concern. Current systems may not offer robust protection against data breaches or unauthorized access. Data loss due to various reasons, including hardware failures or disasters, can be a significant issue for traditional data storage. Additionally, multiple copies of certificates might exist in various formats, leading to redundancy. A blockchain solution can ensure data persistence and reduce redundancy. Managing, issuing, and verifying certificates in traditional systems can be administratively intensive. Blockchain can streamline these processes through automation, potentially reducing costs and the time required for certificate-related tasks. the existing problems in transparent education data management highlight the need for a secure, efficient, and transparent system that leverages blockchain technology. Such a system can tackle issues related to fraud, verification, data security, and access while improving the overall integrity of educational data.

2.2 References

1)https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

2)https://www.mural.co/templates/empathy-map-canvas

3)https://faucet.polygon.technology/

4)https://app.mural.co/t/project1551/m/project1551/1698215591656/6d03fda4c0219ed676d60dd0a7c5956ea07415c8?sender=u2c629690e8195d281d960306

2.3 Problem Statement Definition

In the current landscape of education data management, there exist significant challenges related to the security, accessibility, and authenticity of digital certificates for students. Traditional methods of storing and verifying educational records, including paper-based certificates and even some digital repositories, are prone to issues such as certificate fraud, data tampering, and inefficient verification processes. Additionally, centralized data storage solutions present vulnerabilities to data breaches, and the lack of standardized international recognition of qualifications hampers the mobility of students and professionals.To address these challenges, there is a need for a robust and innovative solution that leverages blockchain technology to create a transparent, secure, and decentralized system for the storage and management of digital certificates. This solution should offer the capability to add certificate details to a blockchain ledger and query these details efficiently. The current systems lack the security measures necessary to prevent certificate fraud and data tampering. The solution should ensure the integrity and authenticity of digital certificates. The process of verifying educational certificates is often slow and resource-intensive. The solution should streamline and automate this verification process, making it more efficient for educational institutions, employers, and individuals.

3.IDEATION & PROPOSED SOLUTION

Ideation:

The primary concept is to utilize blockchain technology as the foundational infrastructure for managing educational certificates. Blockchain's decentralized and immutable ledger will ensure the integrity and trustworthiness of certificate data. Establish a network of nodes representing educational institutions, employers, and verification agencies. These nodes collectively maintain the blockchain, ensuring that there is no central point of failure or control. Implement robust cryptographic techniques to secure the digital certificates. Each certificate's data will be encrypted and only accessible to authorized parties. This ensures data privacy and minimizes the risk of fraud. Employ smart contracts to automate certificate verification processes. These self-executing contracts will enable instant and accurate verification of certificates, reducing administrative overhead.

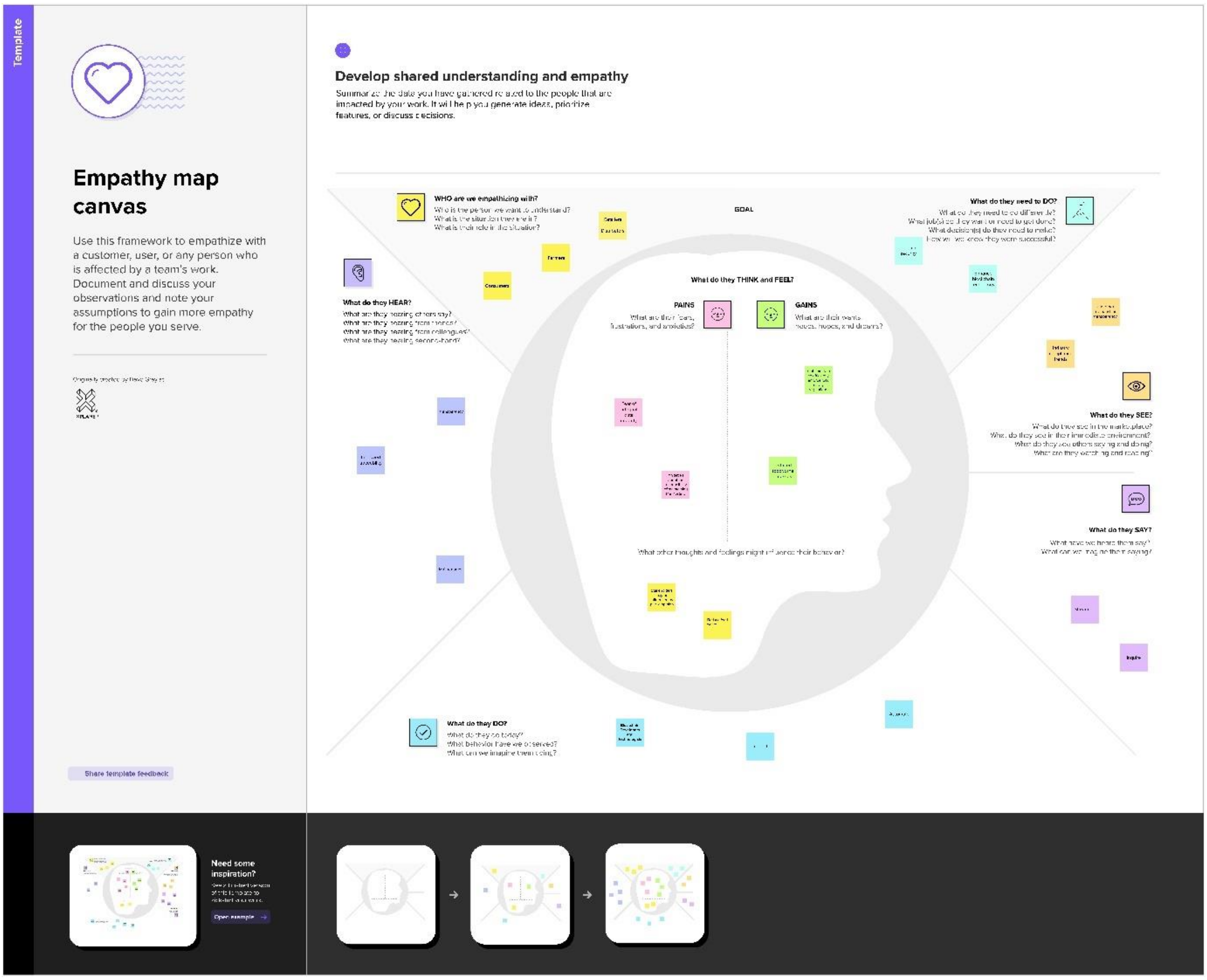
Proposed Solution:

Implement a blockchain network that serves as the backbone of the system. This network will consist of nodes operated by educational institutions, employers, and verification agencies. All certificate data will be encrypted before being added to the blockchain. Access to this data will be restricted to authorized parties using cryptographic keys, ensuring privacy and security. Smart contracts will be employed to automate the verification process. When a certificate is presented for verification, the smart contract will execute and confirm its authenticity, expediting the process. Create a user-friendly web portal for educational institutions to easily input certificate details into the blockchain. A separate portal will allow employers and verification agencies to query and verify certificates . Develop standardized digital certificate formats and metadata to ensure uniformity and compatibility across institutions and industries.

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

Transparent Education Data Management



Empathy map canvas

Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Originally created by Dave Gray at XPLANE

Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.

WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

GOAL

What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

GAINS
What are their wants, needs, hopes, and dreams?

What other thoughts and feelings might influence their behavior?

What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?
What have we heard them say?
What can we imagine them saying?

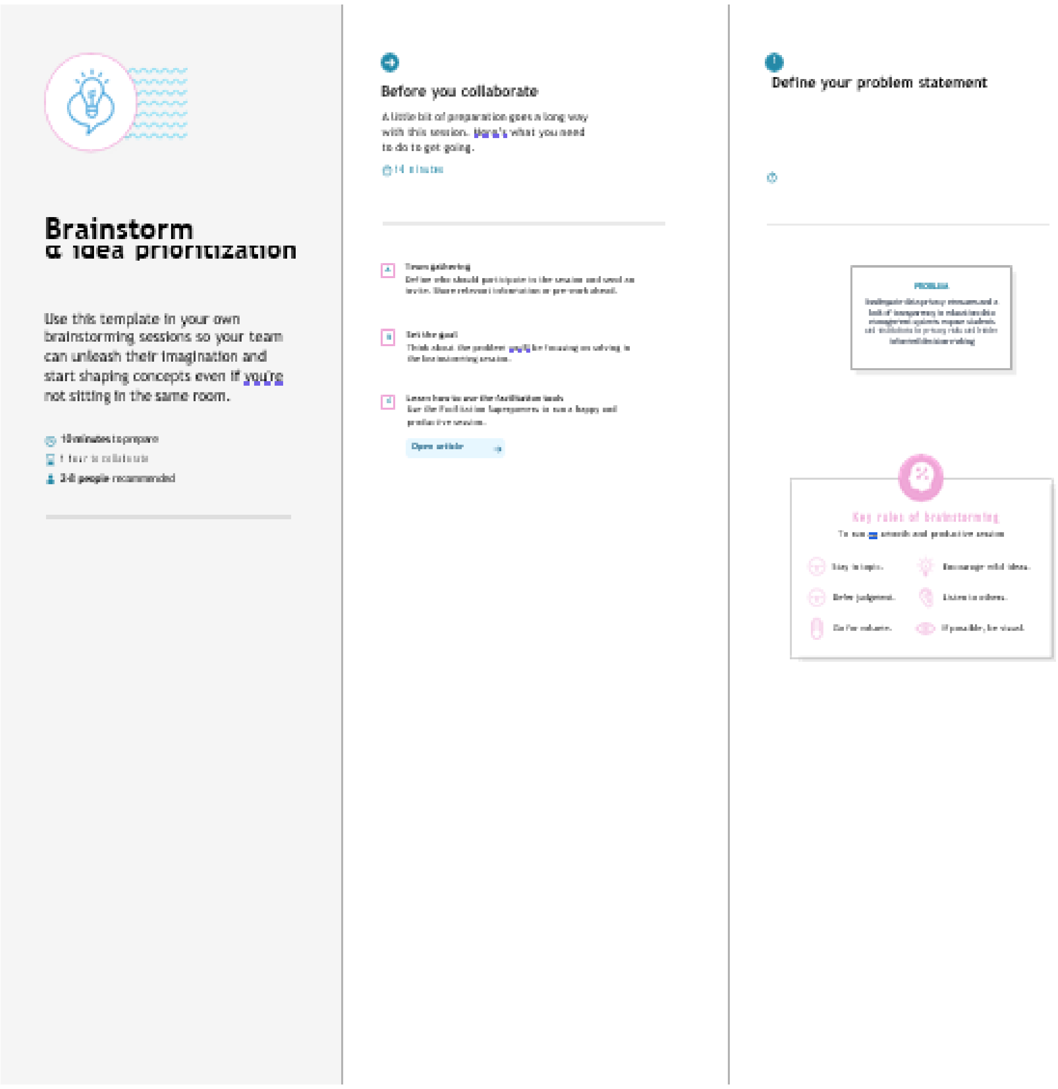
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

9

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants areencouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

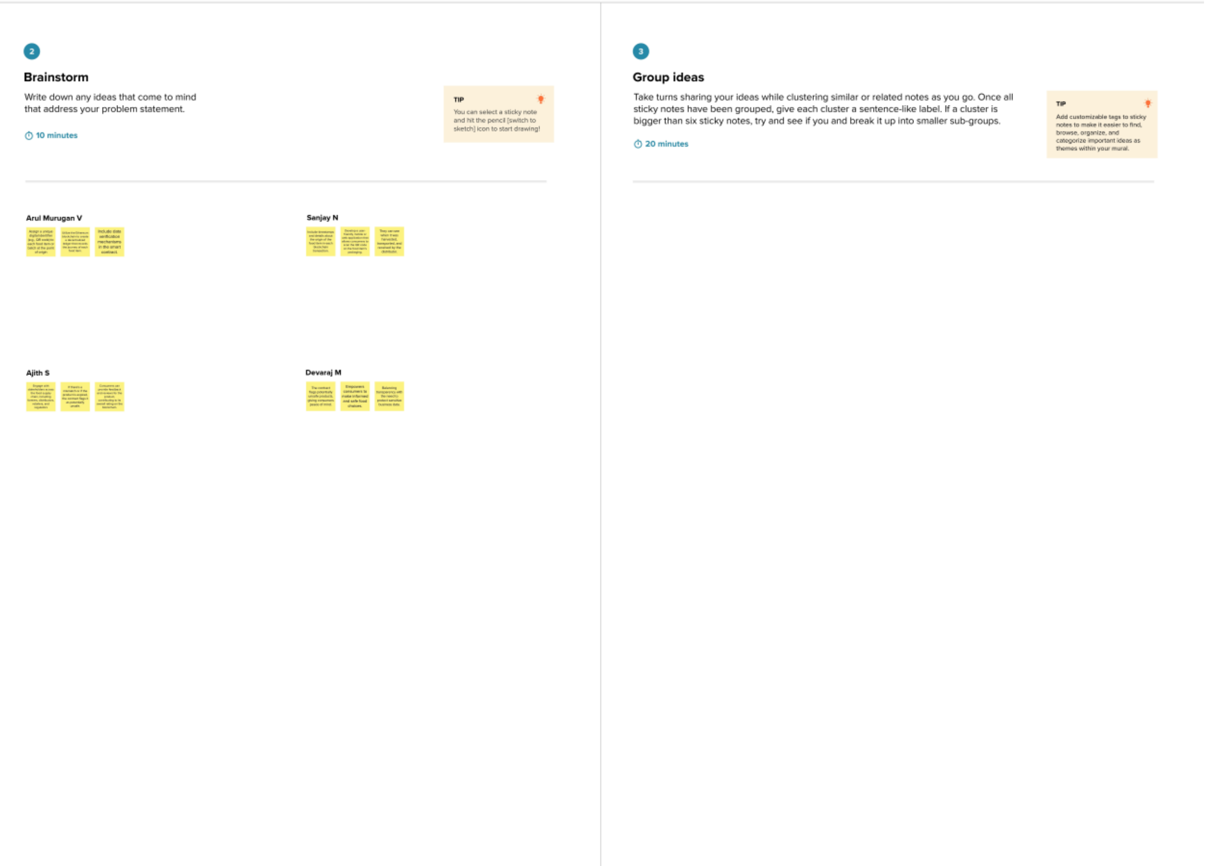
Define your problem statement

PROBLEM
[illegible]

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Arul Murugan V

- Assign a unique digital identifier (e.g., QR code) to each food item or batch at the point of origin.
- Utilize the Ethereum blockchain to create a decentralized ledger that records the journey of each food item.
- Include data verification mechanisms in the smart contract.

Sanjay N

- Include timestamps and details about the origin of the food item in each blockchain transaction.
- Develop a user-friendly mobile or web application that allows consumers to scan the QR code on the food item's packaging.
- They can see when it was harvested, transported, and received by the distributor.

Ajith S

- Engage with stakeholders across the food supply chain, including farmers, distributors, retailers, and regulators.
- If there's a mismatch or if the product is expired, the contract flags it as potentially unsafe.
- Consumers can provide feedback and reviews for the product, contributing to its overall rating on the blockchain.

Devaraj M

- The contract flags potentially unsafe products, giving consumers peace of mind.
- Empowers consumers to make informed and safe food choices.
- Balancing transparency with the need to protect sensitive business data.

3

Group ideas

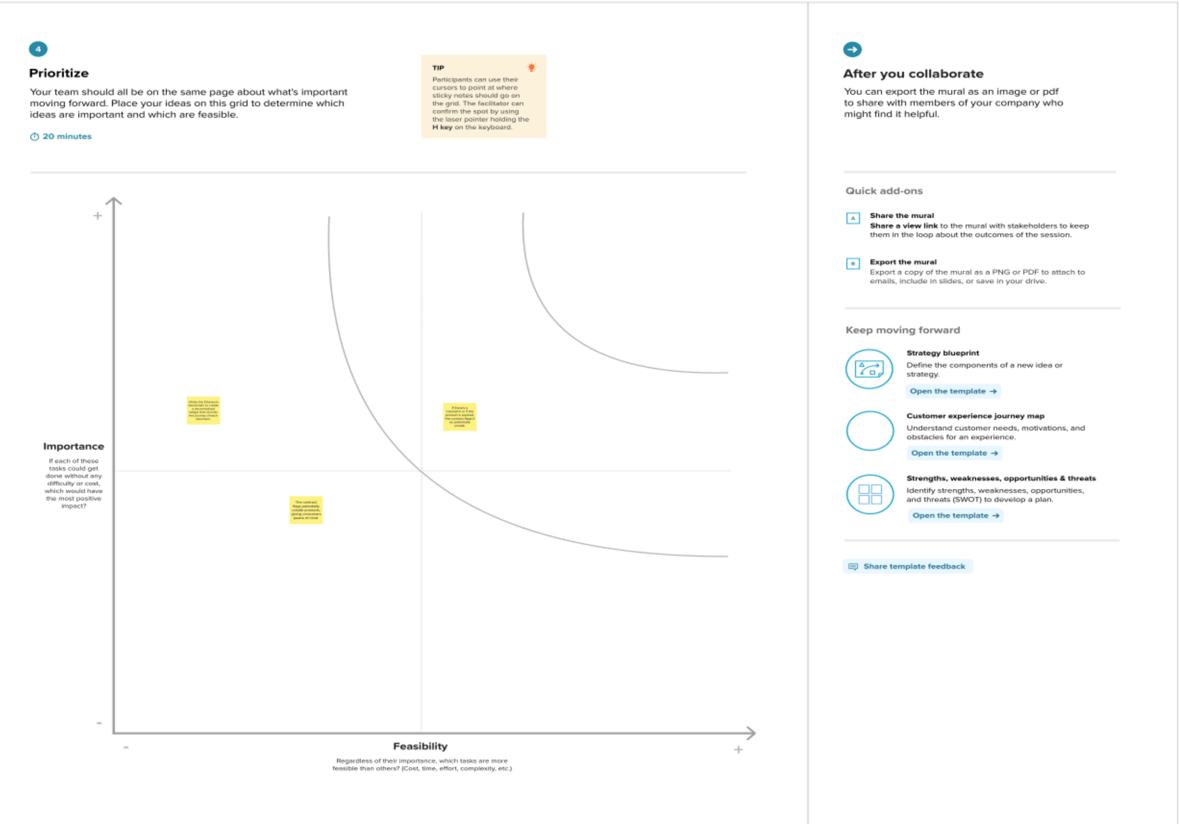
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

- Utilize the Ethereum blockchain to create a decentralized ledger that records the journey of each food item.
- If there's a mismatch or if the product is expired, the contract flags it as potentially unsafe.
- The contract flags potentially unsafe products, giving consumers peace of mind.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

4.REQUIREMENT ANALYSIS

Requirement analysis for transparent education data management in blockchain technology involves identifying the essential features and functions that the system should have to ensure the transparent and secure management of education data. By conducting a thorough requirement analysis and addressing these aspects, you can create a transparent education data management system using blockchain technology that ensures data integrity, security, and accessibility while complying with legal and privacy regulations.

4.1 Functional Requirements

1)Data Storage and Management:

All education data, including academic records, certificates, and transcripts, should be stored as immutable records on the blockchain. Use encryption techniques to ensure the privacy and security of sensitive data.Implement role-based access control to restrict who can read, write, and update data on the blockchain.

2)Data Verification:

Enable employers, educational institutions, and other relevant parties to verify the authenticity of educational credentials stored on the blockchain.Use digital signatures to ensure that data transactions are signed by authorized entities.

3)Interoperability:

Use standardized data formats such as W3C Verifiable Credentials to ensure compatibility with other systems and institutions. Provide APIs to allow external systems to access and interact with the blockchain-based education data.

4)Smart Contracts:

Use smart contracts to automate certain processes, such as verifying credentials or transferring ownership of educational records.Ensure that smart contracts are thoroughly reviewed and audited for security vulnerabilities.

5)User-Friendly Interfaces:

Develop a user-friendly portal for students, educational institutions, and employers to access and manage educational data.Ensure that the system is accessible on mobile devices for easy verification and data access.

4.2 Non Functional Requirements

1)Security:

All data should be stored and transmitted in an encrypted format to protect it from unauthorized access.Implement robust access control mechanisms to prevent unauthorized users from tampering with or viewing sensitive data.Ensure that data stored on the blockchain is truly immutable and resistant to tampering.

2) Reliability and Availability:

Ensure the blockchain network remains available with minimal downtime to guarantee continuous access to education data.Design the system to handle hardware or network failures without data loss or service interruption.

3) Performance:

Define the expected number of transactions per second (TPS) the system must support to meet the demands of educational institutions and employers.Specify acceptable response times for data retrieval and verification processes.The system should be able to scale horizontally and vertically to handle increasing data volumes and user loads.

4)Data Backup and Recovery:

Regularly back up blockchain data to prevent data loss in case of system failures. Develop and maintain a disaster recovery plan to ensure quick system restoration in the event of a catastrophic failure.

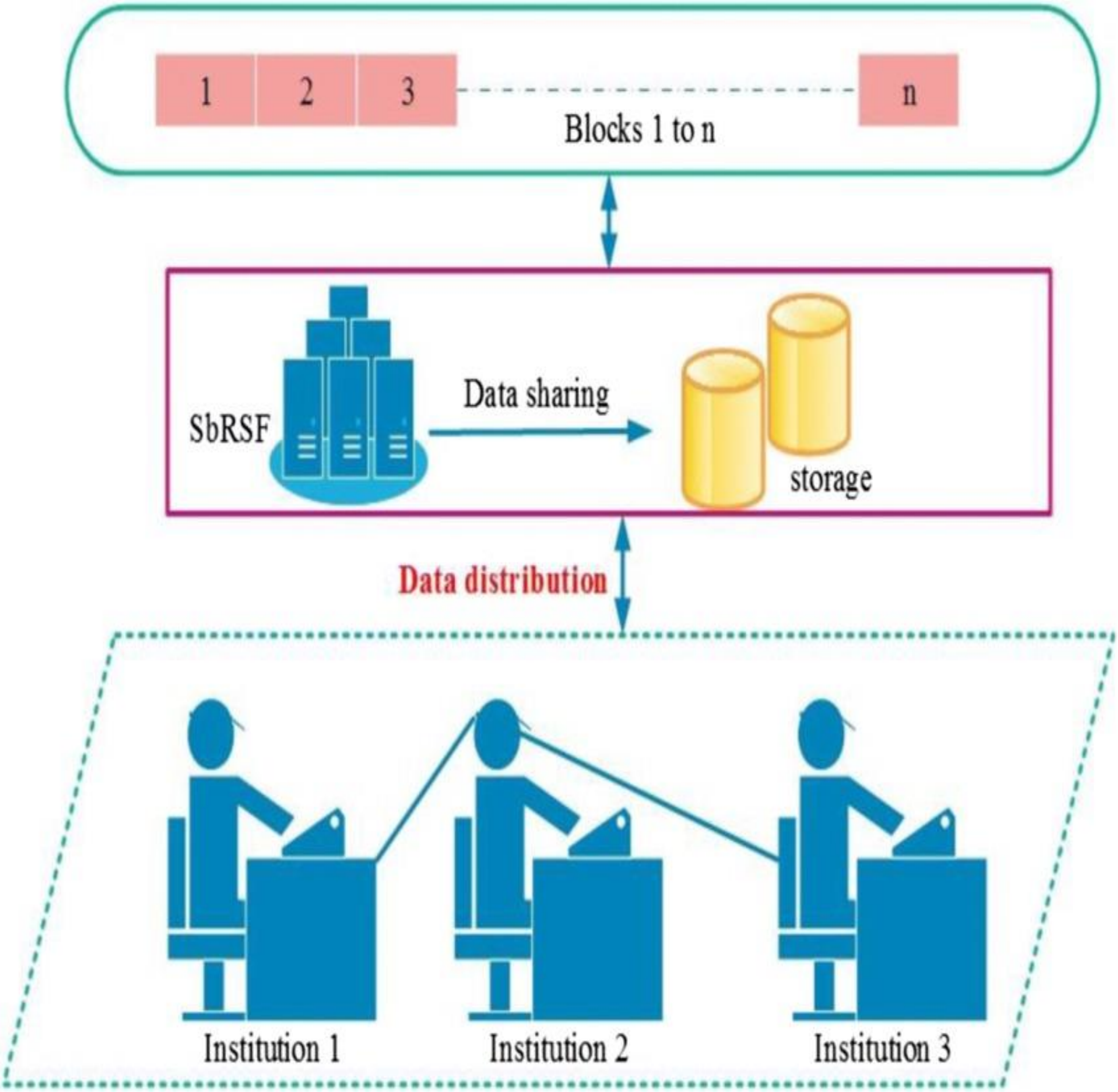
5) Interoperability:

Ensure the system can integrate with existing education management systems and external platforms through well-documented APIs.Follow industry standards for data formats and blockchain interoperability.

5.PROJECT DESIGN

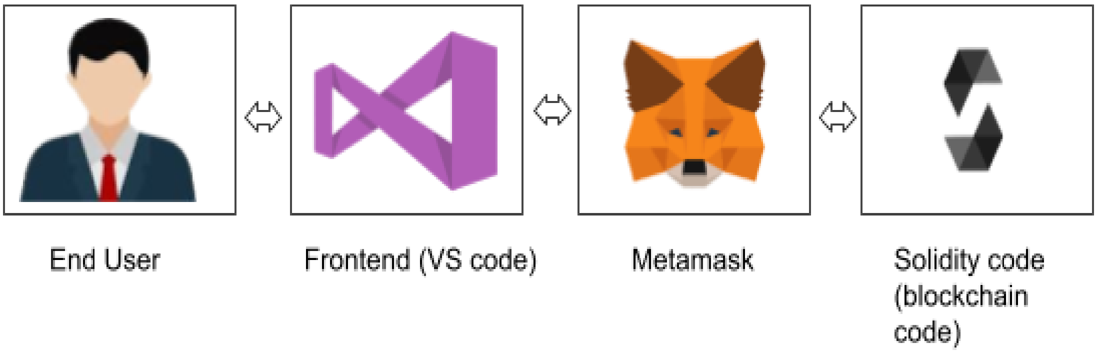
The project aims to implement a blockchain-based system to securely and transparently manage education data, including academic records, certificates, and transcripts, while ensuring data integrity, authenticity, and privacy. The system will serve educational institutions, students, and employers, enabling them to store, access, and verify educational credentials efficiently.

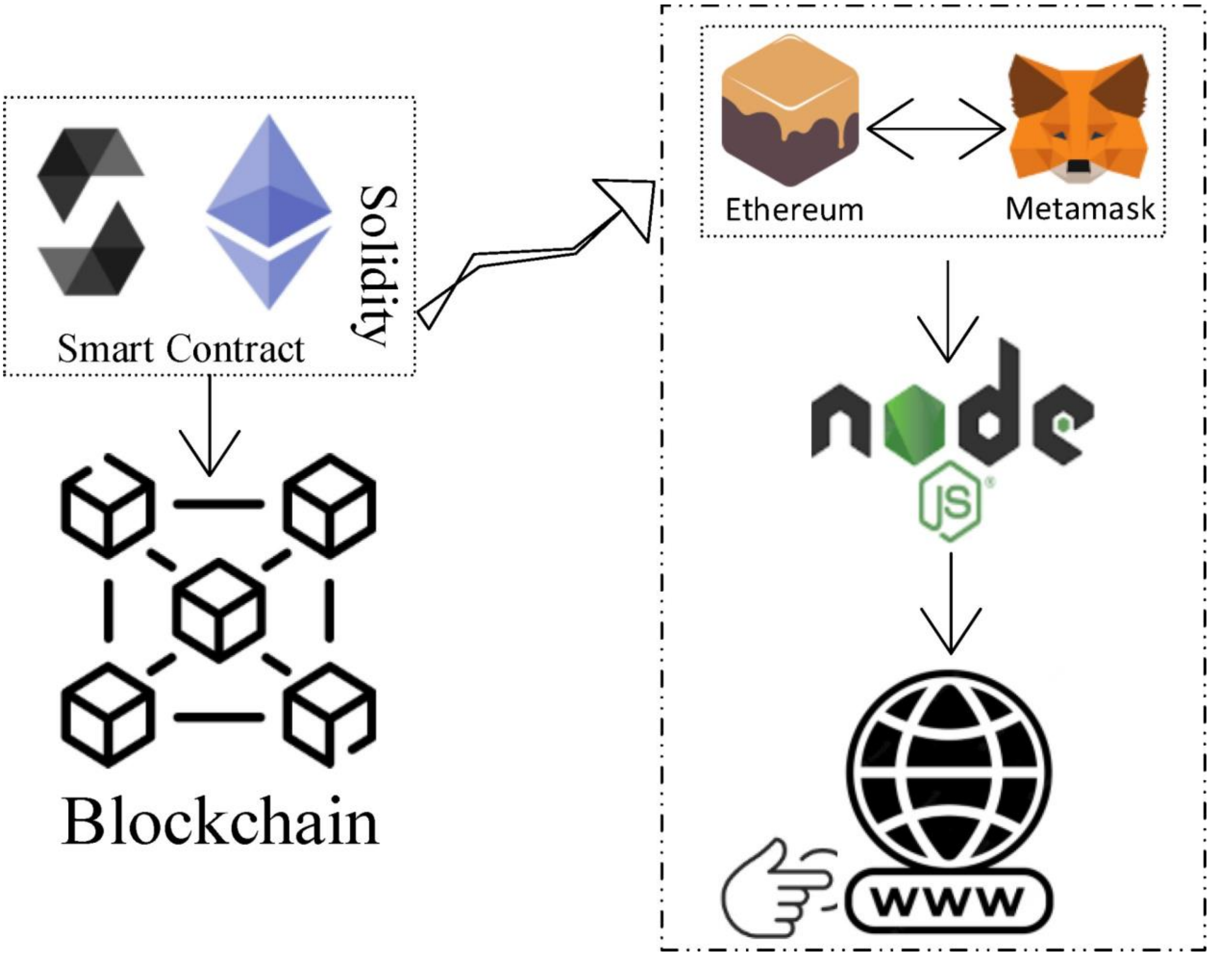
5.1 Data Flow Diagrams & User Stories



5.2 Solution Architecture

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

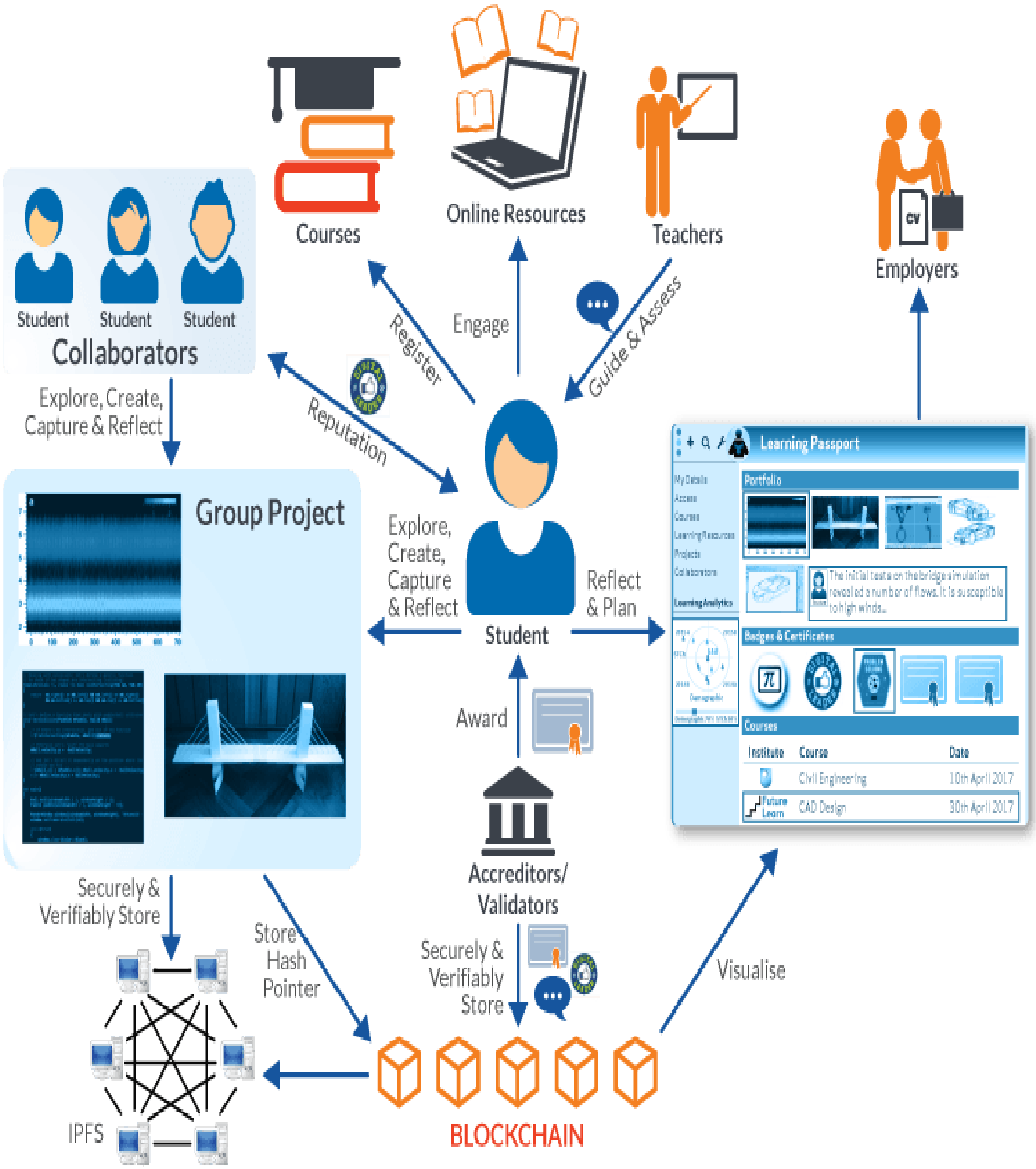




6.PROJECT PLANNING AND SCHEDULING

Project planning and scheduling for transparent education data management is crucial to ensure that educational institutions can effectively collect, store, analyze, and share data while maintaining data security and transparency.

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Tasks:

Research blockchain platforms (Ethereum, Hyperledger) and choose the most suitable one.

Set up a basic blockchain network for testing and development.

Research decentralized storage solutions (IPFS) and evaluate integration options.

Estimated Time: 2 weeks

Sprint 2: Smart Contract Development

Tasks:

Design the data schema for digital certificates.

Develop smart contracts for adding certificates to the blockchain and querying certificate details.

Implement basic functions for certificate validation.

Estimated Time: 3 weeks

Sprint 3: Decentralized Storage Integration

Tasks:

Integrate IPFS or similar decentralized storage for storing actual certificate files.

Ensure secure and efficient file storage and retrieval mechanisms.

Estimated Time: 2 weeks

Sprint 4: User Interface Development

Tasks:

Design user interfaces for students to submit certificates and for verifiers to query certificates.

Implement frontend functionalities for certificate submission and verification.

Estimated Time: 3 weeks

Sprint 5: Security and Testing

Tasks:

Implement encryption and security features to protect data transmission and storage.

Conduct thorough testing, including unit tests, integration tests, and security audits.

Estimated Time: 2 weeks

Sprint 6: Documentation and Deployment

Tasks:

Prepare comprehensive documentation, including system architecture, smart contract details, and user guides.

Deploy the solution on a production-ready blockchain network.

Conduct final testing and address any last-minute issues.

Estimated Time: 2 weeks

Sprint 7: User Acceptance Testing and Refinement

Tasks:

Conduct user acceptance testing (UAT) with stakeholders to ensure the system meets requirements.

Refine the system based on user feedback and address any identified issues.

Estimated Time: 1 week

Sprint 8: Finalization and Deployment

Tasks:

Make final adjustments based on UAT feedback.

Perform a final code review and optimization.

Deploy the solution for production use.

Estimated Time: 1 week

6.3 Sprint Delivery Schedule

Tasks:

Research and select blockchain platform (2 weeks).

Set up a basic blockchain network for testing (2 weeks).

Delivery Date: End of Sprint 1 (4 weeks from project start).

Sprint 2: Smart Contract Development

Tasks:

Design data schema for digital certificates (1 week).

Develop smart contracts for adding and querying certificates (3 weeks).

Delivery Date: End of Sprint 2 (4 weeks from Sprint 1 completion).

Sprint 3: Decentralized Storage Integration

Tasks:

Integrate IPFS for decentralized certificate storage (3 weeks).

Delivery Date: End of Sprint 3 (3 weeks from Sprint 2 completion).

Sprint 4: User Interface Development

Tasks:

Design and implement user interfaces for certificate submission and verification (4 weeks).

Delivery Date: End of Sprint 4 (4 weeks from Sprint 3 completion).

Sprint 5: Security and Testing

Tasks:

Implement encryption and security features (2 weeks).

Conduct testing, including unit tests and integration tests (2 weeks).

Delivery Date: End of Sprint 5 (4 weeks from Sprint 4 completion).

Sprint 6: Documentation and Deployment Preparation

Tasks:

Prepare comprehensive documentation (2 weeks).

Conduct deployment preparations and final optimizations (2 weeks).

Delivery Date: End of Sprint 6 (4 weeks from Sprint 5 completion).

Sprint 7: User Acceptance Testing and Refinement

Tasks:

Conduct user acceptance testing (1 week).

Refine the system based on user feedback (1 week).

Delivery Date: End of Sprint 7 (2 weeks from Sprint 6 completion).

Sprint 8: Finalization and Deployment

Tasks:

Make final adjustments based on UAT feedback (1 week).

Perform final code review and optimizations (1 week).

Deploy the solution for production use (1 week).

Delivery Date: End of Sprint 8 (3 weeks from Sprint 7 completion).

7 CODING & SOLUTIONING

Creating a transparent education data management system using blockchain technology involves various coding and solutioning steps. Below is a high-level outline of the process along with some coding examples to illustrate key concepts. Keep in mind that actual implementation may vary depending on the chosen blockchain platform and programming language. In this example, we'll use Ethereum and Solidity as an example, but other blockchain platforms and languages are also valid options.

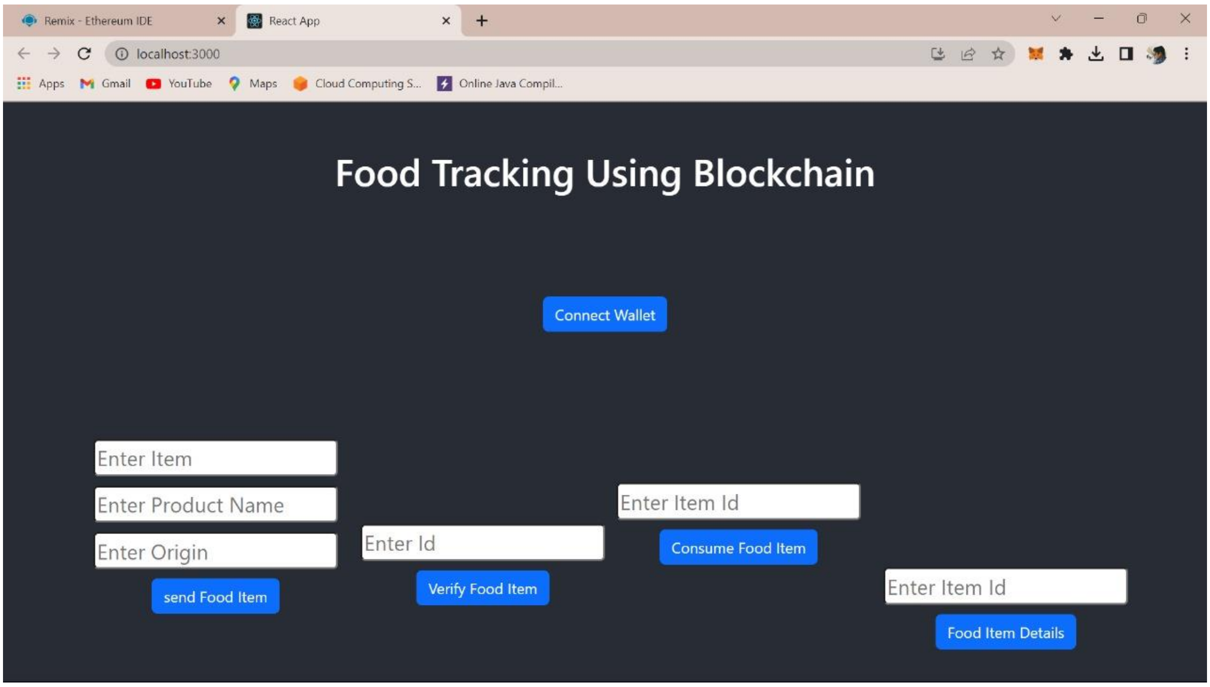
7.1 Feature

```
pragma solidity ^0.8.0;

contract FoodTrackingSystem {

    struct FoodItem {
        uint itemId;
        string itemName;
        address producer;
        address distributor;
        address retailer;
        address consumer;
        bool isAtProducer;
        bool isAtDistributor;
        bool isAtRetailer;
        bool isPurchased;
    }

    mapping(uint => FoodItem) public foodItems;
    uint public itemCount;

    event Produced(uint itemId, string itemName, address producer);
    event Distributed(uint itemId, address distributor);
    event Sold(uint itemId, address retailer);
    event Purchased(uint itemId, address consumer);

    function produceItem(string memory _itemName) public {
        itemCount++;
        foodItems[itemCount] = FoodItem(itemCount, _itemName, msg.sender,
address(0), address(0), address(0), true, false, false, false);
        emit Produced(itemCount, _itemName, msg.sender);
    }

    function distributeItem(uint _itemId) public {
        require(foodItems[_itemId].isAtProducer, "Item not at producer");
        foodItems[_itemId].isAtProducer = false;
        foodItems[_itemId].isAtDistributor = true;
        foodItems[_itemId].distributor = msg.sender;
        emit Distributed(_itemId, msg.sender);
    }

    function sellItem(uint _itemId) public {
        require(foodItems[_itemId].isAtDistributor, "Item not at distributor");
        foodItems[_itemId].isAtDistributor = false;
        foodItems[_itemId].isAtRetailer = true;
        foodItems[_itemId].retailer = msg.sender;
        emit Sold(_itemId, msg.sender);
    }

    function purchaseItem(uint _itemId) public {
        require(foodItems[_itemId].isAtRetailer, "Item not at retailer");
        foodItems[_itemId].isAtRetailer = false;
        foodItems[_itemId].isPurchased = true;
        foodItems[_itemId].consumer = msg.sender;
        emit Purchased(_itemId, msg.sender);
    }
}
```

7.2 Feature

Transparent education data management in blockchain technology should provide a secure and reliable system for recording, verifying, and sharing educational data while ensuring data integrity and privacy. Utilize a distributed ledger to store educational records across a network of nodes, preventing a single point of failure and enhancing data availability. Ensure that once data is recorded on the blockchain, it cannot be altered or deleted, providing a tamper-proof history of educational achievements.

Allow students and educational institutions to control access to their data, granting and revoking permissions as needed. Enable educational institutions to issue digital credentials, certificates, diplomas, and transcripts in a standardized, blockchain-compatible format. Provide a mechanism for third parties (e.g., employers or other educational institutions) to easily verify the authenticity of issued credentials. Implement strong data encryption and privacy measures to safeguard sensitive information while adhering to relevant data protection regulations.



8 PERFORMANCE TESTING

Performance metrics for transparent education data management in blockchain technology are crucial for assessing the efficiency, effectiveness, and reliability of the system. These metrics help in monitoring, optimizing, and ensuring that the blockchain-based education data management platform functions as expected.

8.1 Performance Metrics

Transaction Throughput:

Measure the number of transactions (e.g., credential issuance, data updates) processed per second. Higher throughput is generally desirable to accommodate a larger user base.

Latency and Confirmation Time:

Monitor the time it takes for a transaction to be confirmed and included in a block on the blockchain. Low latency ensures timely access to data.

Scalability:

Assess the system's ability to handle an increasing volume of data and users without significant degradation in performance.

Network Traffic:

Analyze network traffic and bandwidth usage to ensure it remains within acceptable limits, especially in cases of high transaction volumes.

Storage Usage:

Keep track of blockchain storage utilization to ensure it remains manageable as the system grows.

Node Performance:

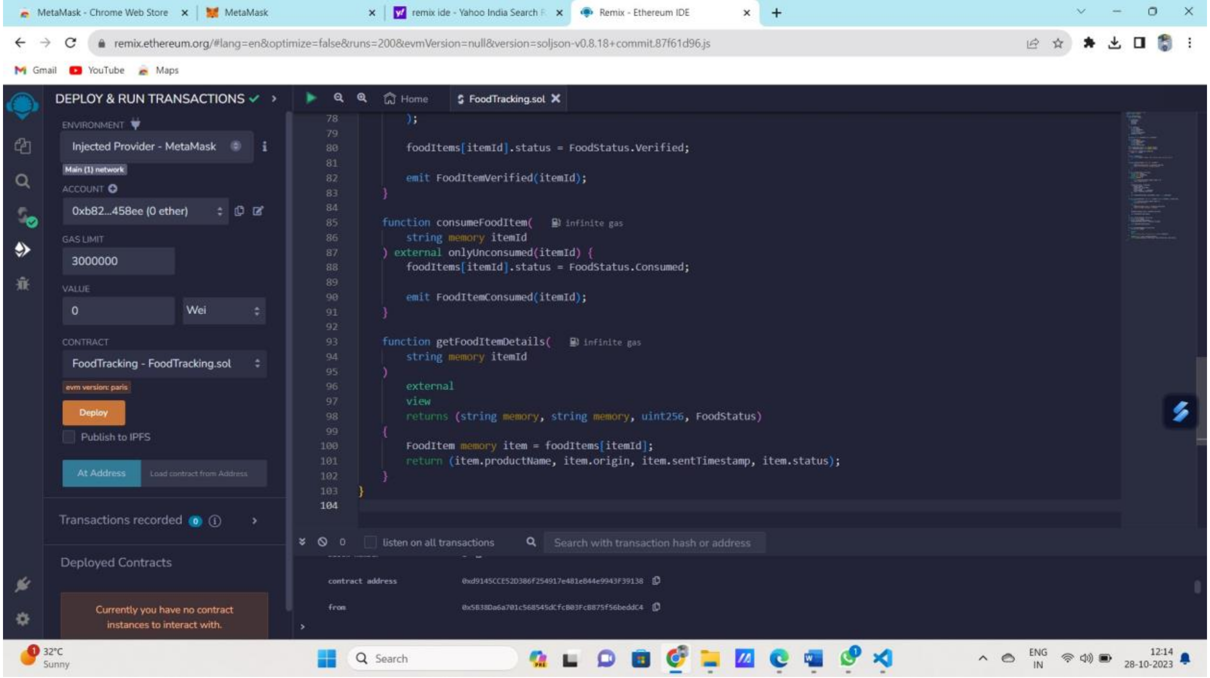
Evaluate the performance of nodes (computers participating in the blockchain network) to identify any bottlenecks or underperforming nodes.

Consensus Overhead:

9 RESULTS

Transparent education data management in blockchain technology can yield a wide range of results and benefits. These results are particularly relevant to students, educational institutions, employers, and other stakeholders involved in the education sector. Blockchain technology ensures that educational records are stored in a tamper-proof and highly secure manner, protecting sensitive data from unauthorized access and tampering. The immutability of blockchain records builds trust among stakeholders by providing a transparent and auditable ledger of educational achievements and qualifications. Blockchain-based credentials are highly resistant to forgery and misrepresentation, reducing the risk of credential fraud and misrepresentation. Employers and educational institutions can quickly and accurately verify the authenticity of credentials, saving time and resources. Students have greater control over their educational data, deciding who can access and verify their records.

9.1 Output Screenshot



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask
Main (1) network

ACCOUNT: 0xb82...458ee (0 ether)

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: FoodTracking - FoodTracking.sol
evm version: paris

Deploy

□ Publish to IPFS

At Address | Load contract from Address

Transactions recorded 0

Deployed Contracts

Currently you have no contract instances to interact with.

FoodTracking.sol

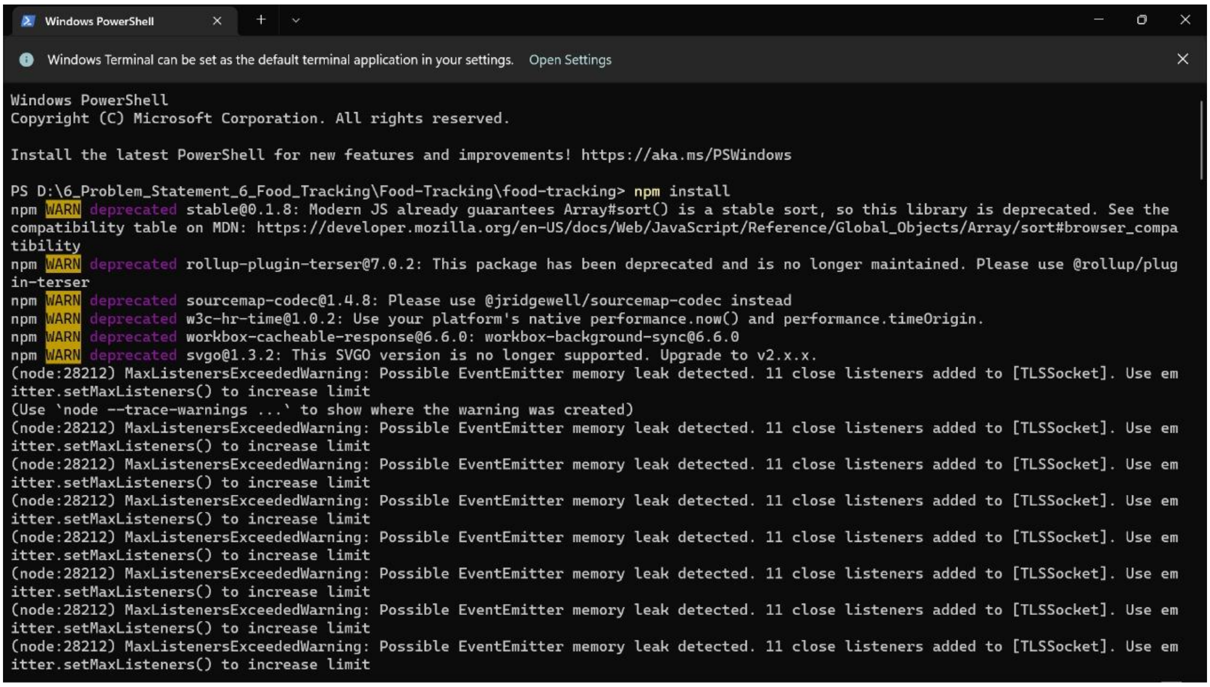
```
        );

        foodItems[itemId].status = FoodStatus.Verified;

        emit FoodItemVerified(itemId);
    }

    function consumeFoodItem(    infinite gas
        string memory itemId
    ) external onlyUnconsumed(itemId) {
        foodItems[itemId].status = FoodStatus.Consumed;

        emit FoodItemConsumed(itemId);
    }

    function getFoodItemDetails(    infinite gas
        string memory itemId
    )
        external
        view
        returns (string memory, string memory, uint256, FoodStatus)
    {
        FoodItem memory item = foodItems[itemId];
        return (item.productName, item.origin, item.sentTimestamp, item.status);
    }
}
```

□ listen on all transactions

```
contract address    0xd9145CCE52D386f254917e481eB44e9943F39138
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
```

MetaMask - Chrome Web Store
MetaMask
remix ide - Yahoo India Search
Remix - Ethereum IDE
Gmail
YouTube
Maps
Home
Search with transaction hash or address
32°C
Sunny
Search
ENG
IN
12:14
28-10-2023

Creating smart contract

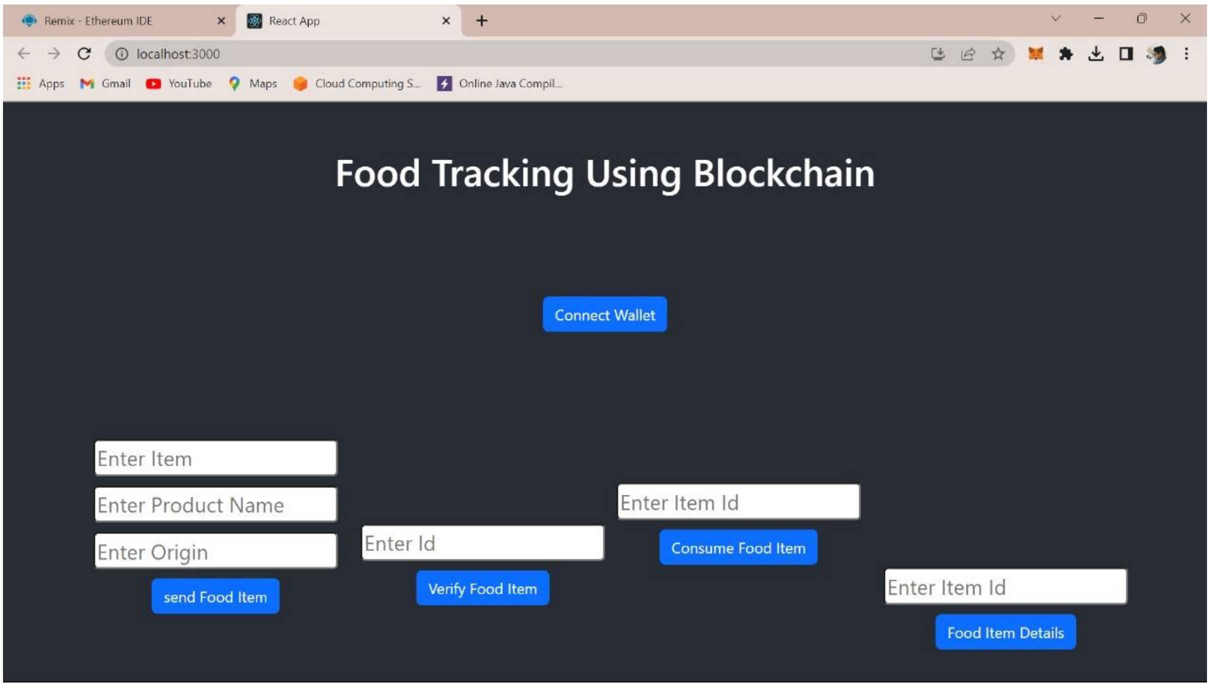


Windows PowerShell

Windows Terminal can be set as the default terminal application in your settings. Open Settings

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\6_Problem_Statement_6_Food_Tracking\Food-Tracking\food-tracking> npm install
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the
compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compa
tibility
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plug
in-terser
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(Use `node --trace-warnings ...` to show where the warning was created)
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
(node:28212) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 close listeners added to [TLSSocket]. Use em
itter.setMaxListeners() to increase limit
```

Installing packages



Output Screenshot

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1) Educational records are stored on a decentralized and tamper-proof ledger, safeguarding them from unauthorized access, manipulation, or loss.
2) Students have greater control over their educational data, deciding who can access and verify their records, thus enhancing privacy.
3) Blockchain-based credentials are highly resistant to forgery and misrepresentation, reducing the risk of credential fraud.
4) Educational records can be accessed from anywhere in the world, making it easier for international students and employers.
5) Access to transparent data enables educational institutions to conduct better data analytics for continuous improvement and personalization.

DISADVANTAGES:

1) Blockchain technology is complex and requires specialized expertise. Developing, maintaining, and securing the system can be challenging and may require significant resources.
2) Some blockchain platforms face scalability limitations, leading to slower transaction processing times as the network grows.
3) Transaction fees associated with blockchain can be high, especially during network congestion, potentially making it expensive for students and institutions to use the system
4) While blockchain provides data security, it may also create challenges related to data privacy and GDPR compliance, particularly if personal information is stored on the blockchain.
5) Widespread adoption of blockchain-based systems may take time due to the need for educational institutions and employers to change their practices and adopt new technology.

11 CONCLUSION

Transparent education data management in blockchain technology holds significant promise for revolutionizing how educational records are stored, managed, and verified. By leveraging the transparency, security, and decentralization of blockchain, educational institutions, students, and employers can benefit in various ways. The advantages include enhanced data security, privacy, and authenticity, streamlined verification processes, and the potential for significant cost savings. It also has the potential to improve data analytics, promote innovation, and facilitate a more collaborative and user-centric educational ecosystem.

However, there are notable challenges and disadvantages to consider, including the technical complexity of blockchain, scalability issues, regulatory uncertainties, and potential barriers to adoption. Overcoming these challenges and maximizing the benefits of blockchain-based education data management requires careful planning, collaboration, and a clear understanding of the specific needs and requirements of the education sector.

In conclusion, transparent education data management in blockchain technology represents a promising paradigm shift in the education sector, offering a path toward greater data security, accessibility, and efficiency. To succeed in implementing such systems, it is essential to address technical and regulatory challenges, provide user education, and foster collaboration among educational institutions, employers, and technology providers. With the right strategies and ongoing commitment, blockchain can help shape a more transparent, secure, and user-centric future for education data management.

12 FUTURE SCOPE

The future scope for food tracking systems is extensive, driven by the increasing demand for transparency, safety, and sustainability in the food supply chain. Innovations in technologies like blockchain, Internet of Things (IoT), artificial intelligence, and data analytics offer significant potential for the evolution of food tracking systems. Blockchain-based solutions, like those built with Solidity in Ethereum, are likely to expand, providing immutable and transparent records of a food item's journey from farm to table. Integration with IoT devices can enable real-time monitoring of environmental conditions, ensuring proper storage and transport, while AI and machine learning can be employed to analyze vast amounts of data for quality control and predictive maintenance. Moreover, the application of smart contracts could revolutionize payment and supply chain logistics, automating transactions upon certain conditions being met. The future also holds promise for consumer-oriented applications, empowering individuals to trace the origin, authenticity, and ethical aspects of the products they consume through easy-to-use interfaces or mobile apps. Collaboration between various stakeholders, standardization, and regulatory support will be crucial for the widespread adoption and success of these advanced food tracking systems, fostering trust and accountability across the global food supply chain.

13 APPENDIX

Source Code

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract FoodTracking {

    address public owner;

    enum FoodStatus {

        Unverified,

        Verified,

        Consumed

    }

    struct FoodItem {

        string itemId;

        string productName;

        string origin;

        uint256 sentTimestamp;

        FoodStatus status;
```

```
}

mapping(string => FoodItem) public foodItems;
event FoodItemSent(
    string indexed itemId,
    string productName,
    string origin,
    uint256 sentTimestamp
);
event FoodItemVerified(string indexed itemId);
event FoodItemConsumed(string indexed itemId);
constructor() {
    owner = msg.sender;
}
modifier onlyOwner() {
    require(msg.sender == owner, "Only contract owner can call this");
    _;
}
modifier onlyUnconsumed(string memory itemId) {
    require(
        foodItems[itemId].status == FoodStatus.Verified,
        "Item is not verified or already consumed"
    );
    _;
}
function sendFoodItem(
    string memory itemId,
    string memory productName,
    string memory origin
) external onlyOwner {
    require(
        bytes(foodItems[itemId].itemId).length == 0,
        "Item already exists"
    );
  foodItems[itemId] = FoodItem({
        itemId: itemId,
        productName: productName,
        origin: origin,
        sentTimestamp: block.timestamp,
        status: FoodStatus.Unverified
    });
  emit FoodItemSent(itemId, productName, origin, block.timestamp);
}
function verifyFoodItem(string memory itemId) external onlyOwner {
    require(
        bytes(foodItems[itemId].itemId).length > 0,
        "Item does not exist"
    );
    require(
        foodItems[itemId].status == FoodStatus.Unverified,
        "Item is already verified or consumed"
    );
    foodItems[itemId].status = FoodStatus.Verified;
    emit FoodItemVerified(itemId);
}
function consumeFoodItem(
    string memory itemId
) external onlyUnconsumed(itemId) {
    foodItems[itemId].status = FoodStatus.Consumed;
    emit FoodItemConsumed(itemId);
}
function getFoodItemDetails(
    string memory itemId
)
    external
    view
```

```
        returns (string memory, string memory, uint256, FoodStatus)
    {
        FoodItem memory item = foodItems[itemId];
        return (item.productName, item.origin, item.sentTimestamp, item.status);
    }
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.
      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.


      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

Connector.js

```
const { ethers } = require("ethers");


const abi = [
 {
  "inputs": [
   {
    "internalType": "string",
    "name": "itemId",
    "type": "string"
   }
  ],
  "name": "consumeFoodItem",
  "outputs": [],
 "stateMutability": "nonpayable",
 "type": "function"
},
{
 "inputs": [],
 "stateMutability": "nonpayable",
 "type": "constructor"
},
{
 "anonymous": false,
 "inputs": [
  {
   "indexed": true,
   "internalType": "string",
   "name": "itemId",
   "type": "string"
  }
 ],
 "name": "FoodItemConsumed",
 "type": "event"
},
{
```

```
"anonymous": false,
"inputs": [
 {
  "indexed": true,
  "internalType": "string",
  "name": "itemId",
  "type": "string"
 },
 {
  "indexed": false,
  "internalType": "string",
  "name": "productName",
  "type": "string"
 },
 {
  "indexed": false,
  "internalType": "string",
  "name": "origin",
  "type": "string"
 },
 {
  "indexed": false,
    "internalType": "uint256",
    "name": "sentTimestamp",
    "type": "uint256"
   }
  ],
  "name": "FoodItemSent",
  "type": "event"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "string",
    "name": "itemId",
    "type": "string"
   }
  ],
  "name": "FoodItemVerified",
  "type": "event"
 },
 {
```

```
 "inputs": [
  {
   "internalType": "string",
   "name": "itemId",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "productName",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "origin",
   "type": "string"
  }
 ],
 "name": "sendFoodItem",
 "outputs": [],
 "stateMutability": "nonpayable",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "itemId",
   "type": "string"
  }
 ],
 "name": "verifyFoodItem",
 "outputs": [],
 "stateMutability": "nonpayable",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  }
 ],
 "name": "foodItems",
"outputs": [
 {
  "internalType": "string",
  "name": "itemId",
  "type": "string"
 },
 {
  "internalType": "string",
  "name": "productName",
  "type": "string"
 },
 {
  "internalType": "string",
  "name": "origin",
  "type": "string"
 },
 {
  "internalType": "uint256",
  "name": "sentTimestamp",
  "type": "uint256"
 },
 {
   "internalType": "enum FoodTracking.FoodStatus",
   "name": "status",
   "type": "uint8"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "itemId",
   "type": "string"
  }
 ],
 "name": "getFoodItemDetails",
 "outputs": [
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  },
  {
   "internalType": "uint256",
   "name": "",
   "type": "uint256"
  },
  {
   "internalType": "enum FoodTracking.FoodStatus",
   "name": "",
   "type": "uint8"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [],
  "name": "owner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 }
]

if (!window.ethereum) {
 alert('Meta Mask Not Found')
 window.open("https://metamask.io/download/")
}

export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "0xd9145CCE52D386f254917e481eB44e9943F39138"
export const contract = new ethers.Contract(address, abi, signer)
import React,{useState} from "react";
import {Button,Container,Row,Col} from 'react-bootstrap';
import 'bootstrap/dist/css/bootstrap.min.css';
import { contract } from "./connector";

function Home() {
  const [handleId, sethandleId] = useState("");
  const [productName, setproductName] = useState("");
  const [origin, setorigin] = useState("");
  const [Id, setId] = useState("");
  const [verifyFood, setverifyFood] = useState("");
  const [verifyFoodDetails, setverifyFoodDetails] = useState("");
  const [Wallet, setWallet] = useState("");



  const handleItemID = (e) => {
    sethandleId(e.target.value)
  }

  const handleProductName = (e) => {
    setproductName(e.target.value)
  }


  const handleOrigin = (e) => {
    setorigin(e.target.value)
  }


  const handleSendFood = async () => {
    try {
      let tx = await contract.sendFoodItem(handleId, productName, origin)
      let txWait = await tx.wait()
      console.log(txWait);


      alert(txWait.transactionHash)
    } catch (error) {
      alert(error)
    }
  }


  const handleVerifyId = async (e) => {
    setId(e.target.value)
  }
```

```
const handleVerifyFood = async () => {
  try {
    let tx = await contract.verifyFoodItem(Id.toString())
    let txWait = await tx.wait()
    console.log(txWait);
    alert(txWait.transactionHash)
  } catch (error) {
    alert(error)
  }

}

const handleConsume = async () => {
  try {
    let tx = await contract.consumeFoodItem(Id.toString())
    let txWait = await tx.wait()
    console.log(txWait);
    alert(txWait.transactionHash)
  } catch (error) {
    alert(error)
  }
```

```
}


const handleFoodItemsDeatils = async()=> {
  try {
    let tx = await contract.getFoodItemDetails(Id)
    let arr = []
    tx.map(e => arr.push(e))


    setverifyFoodDetails(arr)
    // alert(tx)
  } catch (error) {
    alert(error)
  }
}




const handleWallet = async () => {
  if (!window.ethereum) {
    return alert('please install metamask');
  }


  const addr = await window.ethereum.request({
    method: 'eth_requestAccounts',
  });

  setWallet(addr[0])

}

return (
<div>
 <h1 style={{marginTop:"30px", marginBottom:"80px"}}>Food Tracking Using Blockchain</h1>

    {!Wallet ?

      <Button onClick={handleWallet} style={{ marginTop: "30px", marginBottom: "50px" }}>Connect Wallet </Button>
      :
      <p style={{ width: "250px", height: "50px", margin: "auto", marginBottom: "50px", border: '2px solid #2096f3' }}>{Wallet.slice(0, 6)}....{Wallet.slice(-6)}</p>
    }
    <Container style={{ display: "flex",}}>
    <Row style={{marginBottom:"100px",margin:"auto"}}>

    <Col >

     <div>

        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleItemID} type="string" placeholder="Enter Item"
value={handleId} /> <br />


        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleProductName} type="string" placeholder="Enter Product
Name" value={productName} /> <br />


        <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleOrigin} type="string" placeholder="Enter Origin"
value={origin} /><br />

        <Button onClick={handleSendFood} style={{ marginTop: "10px" }}
variant="primary">send Food Item</Button>

     </div>

    </Col>



    <Col >

     <div>

         <input style={{ marginTop: "100px", borderRadius: "5px" }}
onChange={handleVerifyId} type="string" placeholder="Enter Id" value={Id} />
<br />
```

```
          <Button onClick={handleVerifyFood} style={{ marginTop: "10px" }}
variant="primary">Verify Food Item</Button>
      </div>
     </Col>
    </Row>




      <Row style={{marginTop:"100px"}}>
        <Col>
         <div>
          <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleVerifyId} type="string" placeholder="Enter Item Id"
value={Id} /> <br />
          <Button onClick={handleConsume} style={{ marginTop: "10px" }}
variant="primary">Consume Food Item</Button>
         </div>
        </Col>

        <Col>
         <div>
```

```
              <input style={{ marginTop: "100px", borderRadius: "5px" }} onChange={handleVerifyId} type="number" placeholder="Enter Item Id" value={Id} /> <br />

              <Button onClick={handleFoodItemsDeatils} style={{ marginTop: "10px" }} variant="primary">Food Item Details</Button>

              <p>{verifyFoodDetails.toString()}</p>

            </div>

          </Col>

        </Row>

    </Container>


  </div>

  )

}


export default Home;
```

Github:
https://drive.google.com/drive/folders/16n9htJoxOOKm7oX7TF9hw59uYXjy1xY-?usp=drive_link

DemoVideo:

https://drive.google.com/drive/folders/126GwqSWvqyvw3ACOU_Bw_2pRQzwX07pe?usp=drive_link