# Project Report

1. **INTRODUCTION**
    1.1 Project Overview
    1.2 Purpose
2. **LITERATURE SURVEY**
    2.1 Existing problem
    2.2 References
    2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    3.1 Empathy Map Canvas
    3.2 Ideation & Brainstorming
4. **REQUIREMENT ANALYSIS**
    4.1 Functional requirement
    4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    5.1 Data Flow Diagrams & User Stories
    5.2 Solution Architecture
6. **PROJECT PLANNING & SCHEDULING**
    6.1 Technical Architecture
    6.2 Sprint Planning & Estimation
    6.3 Sprint Delivery Schedule
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    7.1 Feature 1
    7.2 Feature 2
    7.3 Database Schema (if Applicable)
8. **PERFORMANCE TESTING**
    8.1 Performace Metrics
9. **RESULTS**
    9.1 Output Screenshots
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**

**12. FUTURE SCOPE**

**13. APPENDIX**

Source Code

GitHub & Project Demo Link

# Transparent Education Data Management

## 1.INTRODUCTION:

The management of education data has long been a complex challenge, marked by issues of data integrity, transparency, and accountability. This project delves into the transformative potential of blockchain technology in addressing these challenges. Blockchain, originally devised as the underlying technology for cryptocurrencies, offers a decentralized and secure digital ledger system that records transactions across a network. This project aims to showcase how blockchain can reshape education data management, providing a secure and tamper-proof environment that ensures data accuracy and authenticity. [1].

The central objective of this project is to illustrate how blockchain can enhance transparency and accountability within the education sector. By adopting blockchain technology, education-related transactions and records can be securely stored and accessed by all stakeholders, ranging from educational institutions to employers and regulatory bodies.[2] This real-time access to verified and immutable academic records not only simplifies processes but also accelerates credential verification, reduces fraud, and promotes accountability throughout the educational journey..[3].

The education sector is undergoing a digital transformation, with an increasing reliance on technology to streamline administrative processes and enhance learning experiences. One crucial aspect of this transformation is the management and security of education data.[4] Traditional data management systems often suffer from issues related to data integrity, security, and transparency. To address these challenges, many educational institutions and organizations are turning to blockchain technology.[5].

Blockchain technology, renowned for its decentralized and immutable ledger, offers a promising solution to the issues plaguing education data management. This innovative approach can bring much-needed transparency, security, and efficiency to the educational data ecosystem.[6].

## 1.1.Project Overview

The "Transparent Education Data Management Using Blockchain" project aims to revolutionize the education sector by introducing blockchain technology to enhance data security, transparency, and efficiency in education data management. It seeks to provide a secure and transparent system for storing and sharing educational records and credentials, thereby benefiting students, educational institutions, and various stakeholders.

**Key Components:**

**Enhanced Data Security:** Implement a blockchain-based system to enhance the security of educational records and data. Blockchain's decentralized and immutable ledger ensures data integrity and protects against unauthorized access or data breaches.

**Transparency and Verification:** Develop a transparent ecosystem that allows students, educators, and authorized parties to easily access and verify educational records and credentials. This transparency helps in combating issues like credential fraud and misrepresentation.

**Efficient Administrative Processes:** Utilize smart contracts on the blockchain to automate administrative tasks such as transcript requests, enrollment verification, and credential validation. This automation reduces bureaucratic overhead, enhances efficiency, and ensures timely responses to requests.

**Interoperability:**Establish standardized protocols and formats for educational data stored on the blockchain, ensuring that data can be seamlessly shared and verified across different educational institutions and platforms. This promotes data consistency and ease of transfer.

**User Empowerment:** Provide students with greater control over their educational records and credentials by allowing them to securely share this data with potential employers or other educational institutions. This empowerment enhances students' ownership of their education-related data.

**Privacy and Compliance:** Ensure that the project complies with relevant data protection regulations and best practices to safeguard the privacy and rights of individuals whose data is stored on the blockchain.

**Scalability and Cost-Effectiveness:** Evaluate the scalability of the blockchain solution and its cost-effectiveness in education data management. Identify any challenges and propose solutions to address them, considering the long-term sustainability of the system.

**Research and Development:** Continuously assess the evolving landscape of blockchain technology and its applicability to education data management. Engage in research and development activities to refine and improve the system.

## 1.2.Purpose:

The primary purpose of the "Transparent Education Data Management Using Blockchain" project is to revolutionize education data management by integrating blockchain technology. The project aims to:

Enhance Data Security: Implement a robust, decentralized, and immutable blockchain system to significantly improve the security and integrity of educational records and data. This enhances protection against data breaches and unauthorized access.

Promote Transparency: Create a transparent and accessible data ecosystem, allowing students, educators, and authorized stakeholders to easily access, verify, and trust educational records and credentials. This transparency helps combat issues such as credential fraud and misrepresentation.

Streamline Administrative Processes: Utilize smart contracts to automate time-consuming administrative tasks, such as transcript requests, enrollment verification, and credential validation. This automation reduces administrative overhead, enhances efficiency, and ensures rapid responses to requests.

Ensure Interoperability- Establish standardized data protocols and formats for educational records stored on the blockchain, enabling seamless sharing and verification of data across various educational institutions and platforms. This promotes data consistency and portability.

Empower Users:Empower students with greater control over their educational records and credentials, allowing them to securely share this information with prospective employers or other educational institutions. This empowerment increases students' ownership of their educational data.

Privacy and Compliance: Ensure that the project adheres to relevant data protection regulations and best practices, safeguarding individuals' privacy and rights while using blockchain for education data management.

Evaluate Scalability and Cost-Effectiveness: Assess the scalability of the blockchain solution and its long-term cost-effectiveness in education data management, identifying potential challenges and proposing solutions to ensure sustainability.

Research and Development:Engage in ongoing research and development activities to stay abreast of the evolving blockchain technology landscape and improve the system as necessary.

# 2. LITERATURE SURVEY

## 2.1.Existing problem:

**Data Security and Privacy:** Traditional education data management systems are vulnerable to data breaches and unauthorized access. Protecting sensitive information, such as student records and transcripts, is a major concern.

**Credential Verification:**The process of verifying educational credentials can be time-consuming and may lack transparency. This creates challenges for employers, educational institutions, and students.

**Data Fragmentation:**Education data is often stored in multiple, disconnected systems, leading to data fragmentation. This fragmentation can hinder data sharing and result in inconsistencies.

**Data Ownership:**Students often lack control and ownership of their educational data. Blockchain can empower them to control who accesses and validates their records.

**Transcript Authentication:** Verifying the authenticity of transcripts is a common issue. Blockchain can provide a tamper-evident ledger for academic records.

**Standardization :**The absence of standardized formats for educational data can impede interoperability between institutions and platforms.

## 2.2.References:

1. "Blockchain: The Solution for Transparency in Degrees and Certificates"Author: M. Puican,Published in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).This paper discusses the use of blockchain to enhance the transparency and security of educational credentials.

2. "Blockchain in Education"Author: G. Shakya, et al. Published in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), This study explores the potential applications of blockchain in education, focusing on credential verification and data security.

3."Blockchain in Education: Opportunities and Challenges"*Author: L. Breslauer,Published in: EDUCAUSE Review.This article provides an overview of the opportunities and challenges of implementing blockchain in education.

4. "Using Blockchain Technology to Ensure the Transparency and Integrity of Student Graduation Certificates" Author: A. Agbo, et al.
Published in: 2018 IEEE International Conference on Information and Communication Technologies (ICICT).This paper discusses the use of blockchain to ensure the transparency and integrity of student graduation certificates.

5."A New Ecosystem for Educational Credentials: A Progress Report on Blockchain"Author: W. F. Fadel, et al. Published by: The American Academy of Arts and SciencesThis report examines the potential of blockchain technology to create a new ecosystem for educational credentials.

## 2.3.Problem Statement Definition:

### 2.3.1.Problem statement:

In the face of global climate change, there is an increasing need for a reliable, secure, and transparent system to track, verify, and manage climate-related data and assets. Current methods for monitoring carbon emissions, renewable energy production, or carbon credits trading lack transparency, are often subject to fraud, and have limited cross-border compatibility. To address these challenges, the problem statement is to develop a "climate track smart" system using blockchain technology. This system should enable the secure and decentralized tracking of climate-related activities, assets, and data to ensure accuracy, prevent fraud, and facilitate efficient reporting and trading on a global scale.

**Key elements of this problem statement include:**

**Climate Data Tracking:** Designing a system that can accurately track climate-related data, such as carbon emissions, temperature changes, and renewable energy production, in real-time or near-real-time.

**Verification and Transparency:** Ensuring that the system provides transparent, immutable records that can be verified by relevant stakeholders, including governments, organizations, and the public.

**Security and Fraud Prevention:** Implementing robust security measures to prevent fraudulent or unauthorized changes to the data and ensure the integrity of the information.

**Interoperability:** Creating a system that can function across borders and with different types of climate data, enabling global cooperation and consistency.

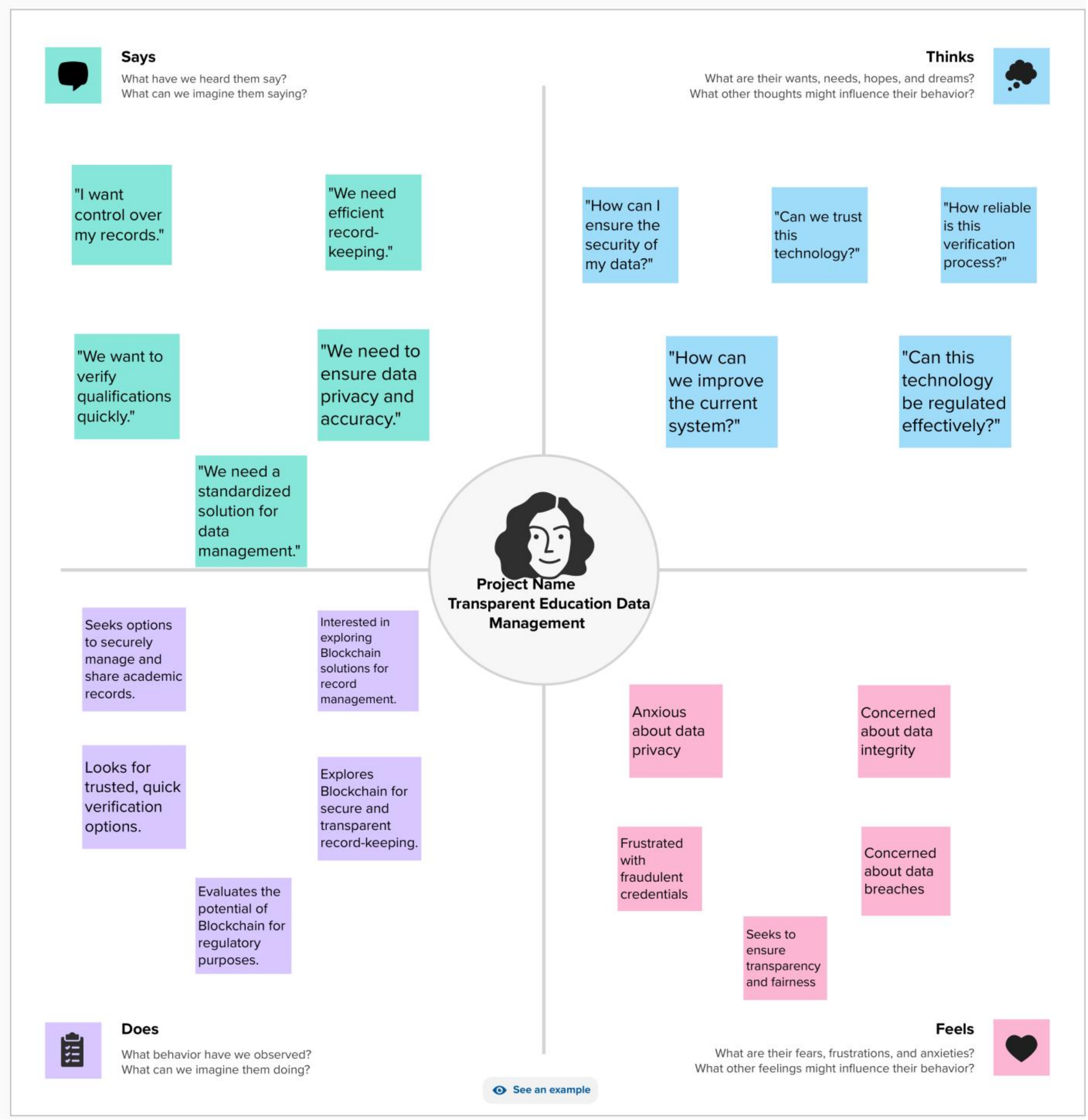
**Efficiency and Automation:** Developing smart contract functionalities or automation to simplify processes such as carbon credit trading, compliance reporting, and data sharing.

| PS | I am (User) | I am Trying to | But | Because | Which Makes me feel |
|---|---|---|---|---|---|
| PS-1 | User | To Record or Review Transparent Education Data Management | **Data Security and Privacy**: | Conventional systems lack robust security measures. | Implement a permissioned Blockchain network with cryptographic encryption to ensure secure and private data storage and transmission. |
| PS-2 | User | To Record or Review Transparent Education Data Management | **Verificationof Academic Credentials** | Manual verification processes are inefficient. | Use Blockchain for transparent, tamper-proof academic credential verification. Institutions can upload data, and verification can be done quickly through a decentralized system. |
| PS-3 | User | To Record or Review Transparent Education Data Management | **Fraudulent Degree Mills**: | Lack of a trusted and centralized authority. | Create a Blockchain-based degree verification system that records and authenticates degrees, making it harder for fraudulent institutions to thrive. |

| | | | | | |
|---|---|---|---|---|---|
| PS-4 | User | To Record or Review Transparent Education Data Management | **Inefficient Record Keeping**: | Paper-based and outdated record-keeping methods. | Implement Blockchain to create a single, immutable ledger for student records, ensuring accuracy and making records easily transferable across institutions. |
| PS-5 | User | To Record or Review Transparent Education Data Management | **Credential Ownership and Control**: . | Institutions typically control and manage credentials. | Develop a Blockchain-based solution that allows students to have ownership and control over their academic records, granting them the ability to share them securely as needed. |

# 3. IDEATION & PROPOSED SOLUTION

## 3.1.Empathy Map Canvas

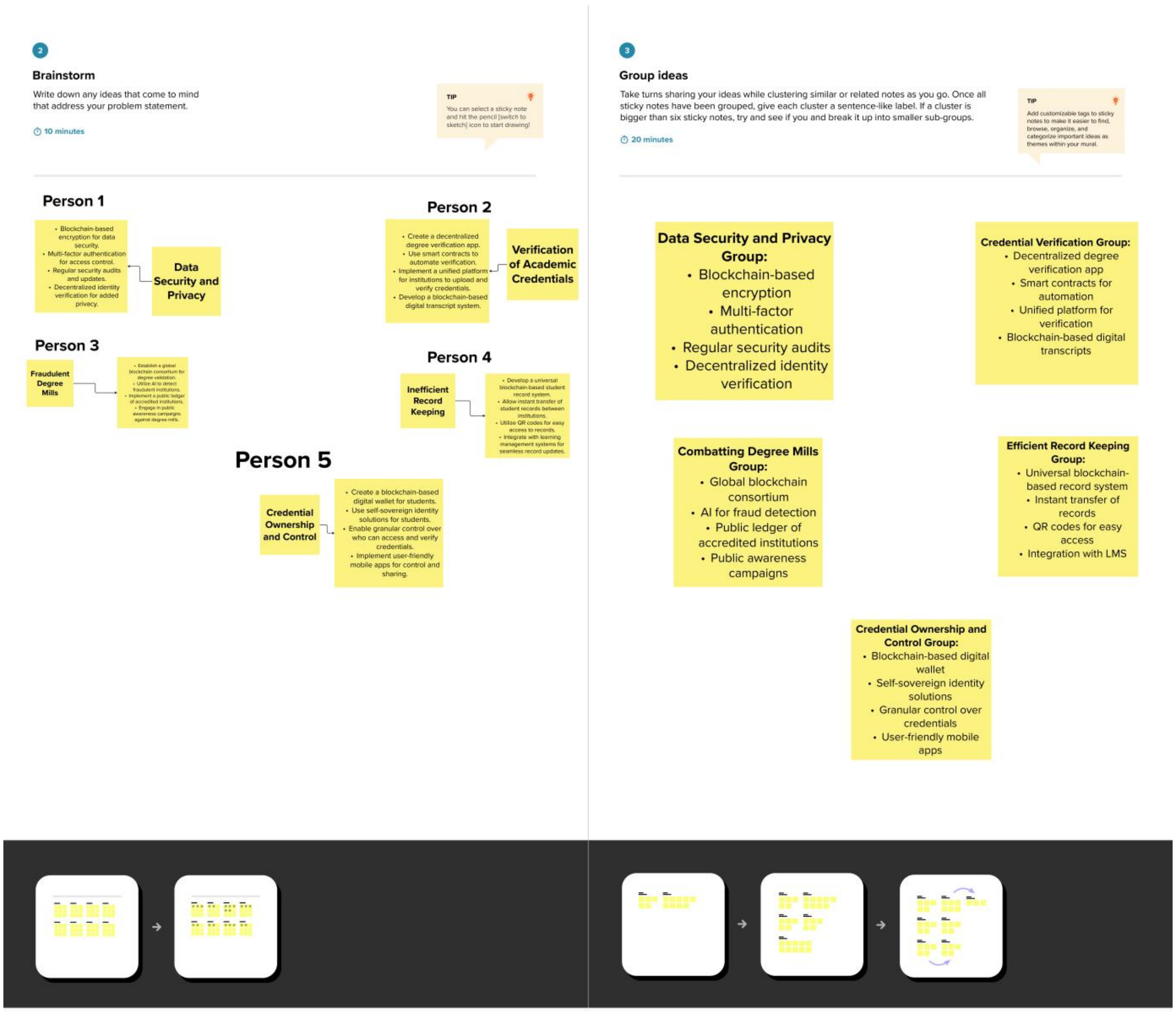


# 3.2.Ideation & Brainstorming

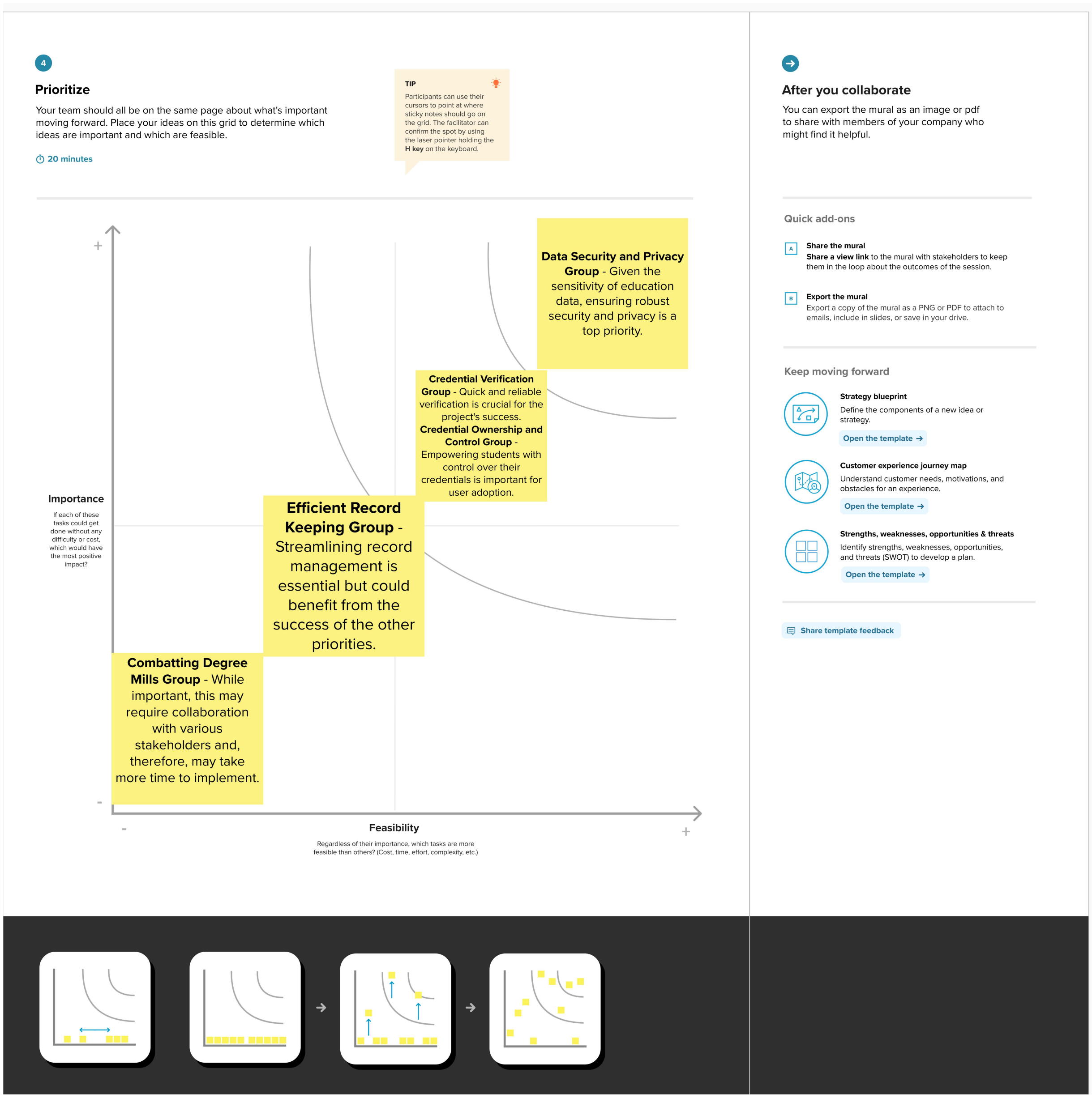
## Brainstorm & Idea Prioritization

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

# Step-2: Brainstorm, Idea Listing and Grouping



# Step-3: Idea Prioritization



# 4. REQUIREMENT ANALYSIS

## 4.1.Functional requirement

Following are the functional requirement of the Proposed Solution.

| FR. no | Functional Requirement(Epic) | Sub-Requirement |
|---|---|---|
| 1 | User Registration and Authentication | Users (students, institutions, employers, regulators) must be able to register and authenticate their identities securely. |
| 2 | Blockchain-Based Record Storage | The system should allow educational institutions to upload and store academic records securely on a blockchain. |
| 3 | Credential Verification | Users should be able to verify academic credentials quickly and reliably through the platform. |
| 4 | Credential Ownership and Control | Students should have control over who can access and verify their academic records and credentials. |

| | | |
|---|---|---|
| 5 | Record Transfer and Sharing | Institutions should be able to transfer student records securely to other educational institutions, and students should be able to share their credentials with employers. |
| 6 | Anti-Fraud Mechanisms | The system should include mechanisms to detect and prevent fraudulent degree mills from gaining credibility. |

### 4.2.Non-Functional Requirement:

Following are the Non-functional requirement of the Proposed Solution.

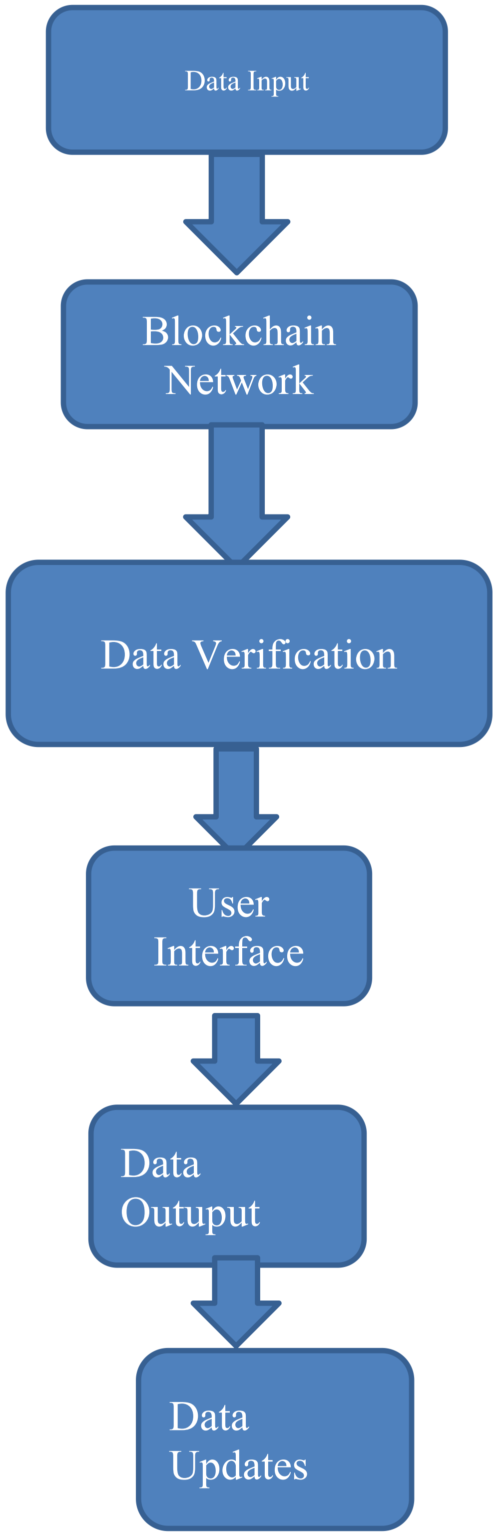
| Non-FR. no | Non-Functional Requirement(Epic) | Sub-Requirement |
|---|---|---|
| 1 | Security | The system must ensure the highest level of data security and privacy to protect sensitive educational records and personal information. |
| 2 | Scalability | The platform should be scalable to accommodate a growing number of users, educational institutions, and records. |
| 3 | Performance | The system must provide fast and reliable academic credential verification and record retrieval. |
| 4 | Usability | The user interface should be intuitive, user-friendly, and accessible to individuals with diverse technical backgrounds. |
| 5 | Compliance and Regulation | The system must adhere to relevant data protection regulations and ensure transparency in line with educational authorities and regulators' requirements |

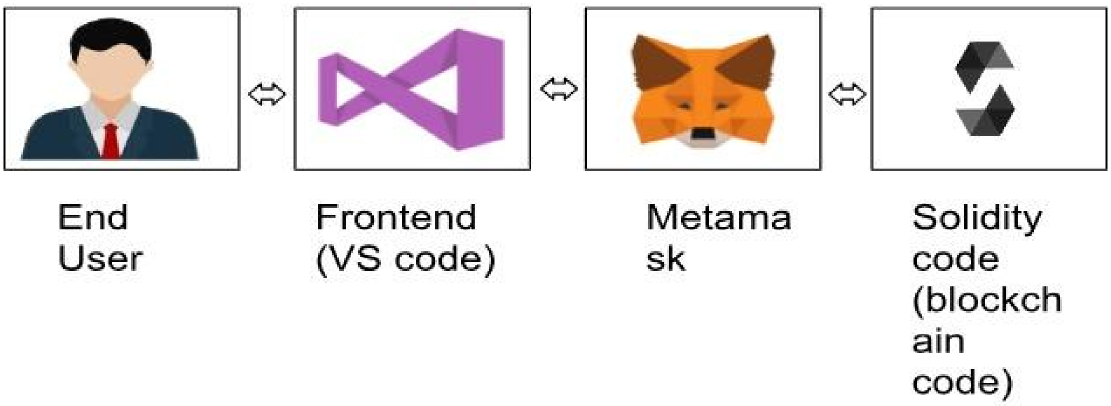
| | | |
|---|---|---|
| 6 | Interoperability | The system should be able to integrate with existing education data systems and provide APIs for third-party applications and services. |

## 5.PROJECT DESIGN

### 5.1.Data Flow Diagrams & User Stories

### 5.1.1Data Flow Diagrams




End User
Frontend (VS code)
Metamask
Solidity code (blockchain code)

### 5.1.2.User Stories:

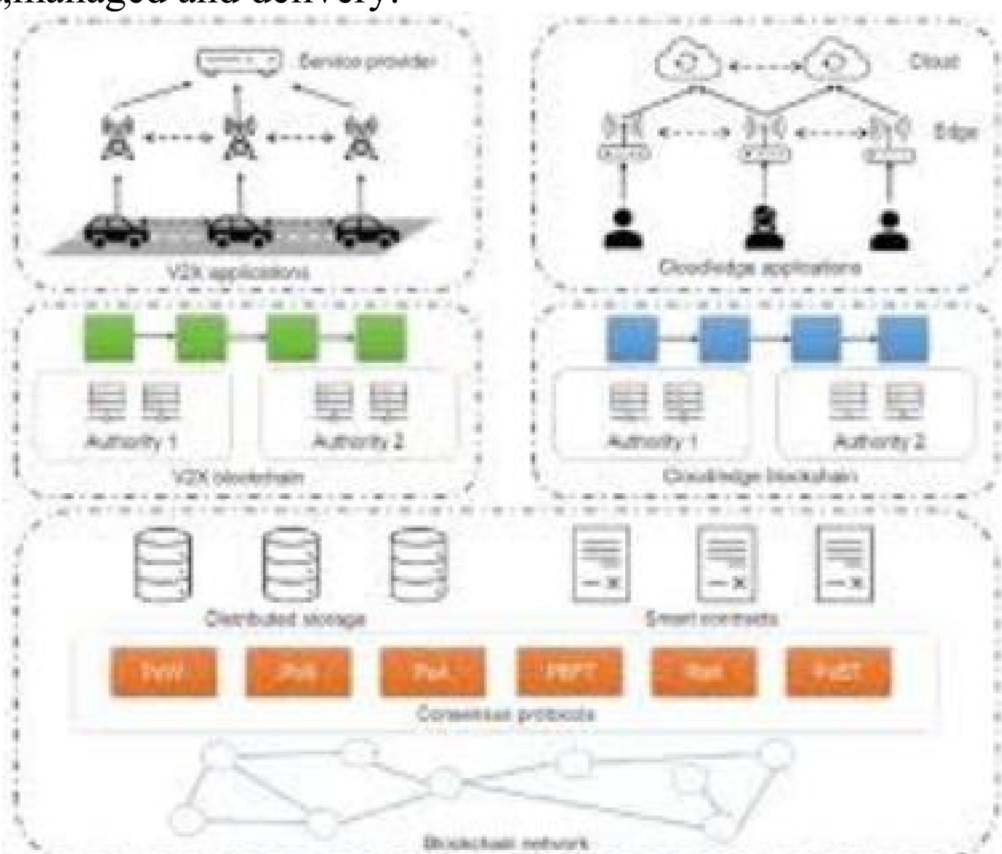
| User Story | Acceptance Criteria |
|---|---|
| **Student**-As a student, I want to securely upload and manage my academic records on the blockchain. | I can register and authenticate my identity securely. I can upload and view my academic records on the blockchain. I have control over who can access and verify my credentials. |
| **Educational Institution**-As an educational institution, I want to upload and store student records on the blockchain. | I can register and authenticate my institution's identity. I can securely upload and store student records on the blockchain. The records are tamper-proof and easily transferable. |
| **Employer**-As an employer, I want to quickly verify the academic credentials of potential employees. | I can register and authenticate my identity. I can enter a candidate's details and quickly verify their academic credentials. The verification process is reliable and secure. |
| **Regulatory Authority**-As a regulatory authority, I want to ensure compliance and transparency in education data management. | -The system complies with data protection regulations and relevant educational standards. I have access to necessary data for oversight and regulation. |
| **System Administrator**-As a system administrator, I want to monitor and maintain the security and performance of the blockchain system. | I can monitor system security and identify potential threats. I can perform regular maintenance and updates. The system's performance is optimized, and issues are addressed promptly. |

## 5.2.Solution Architecture

Solution Architecture is complex process-with may sub -Processes - That Bridges the gap between business problem and Technology solution Its goals are to;

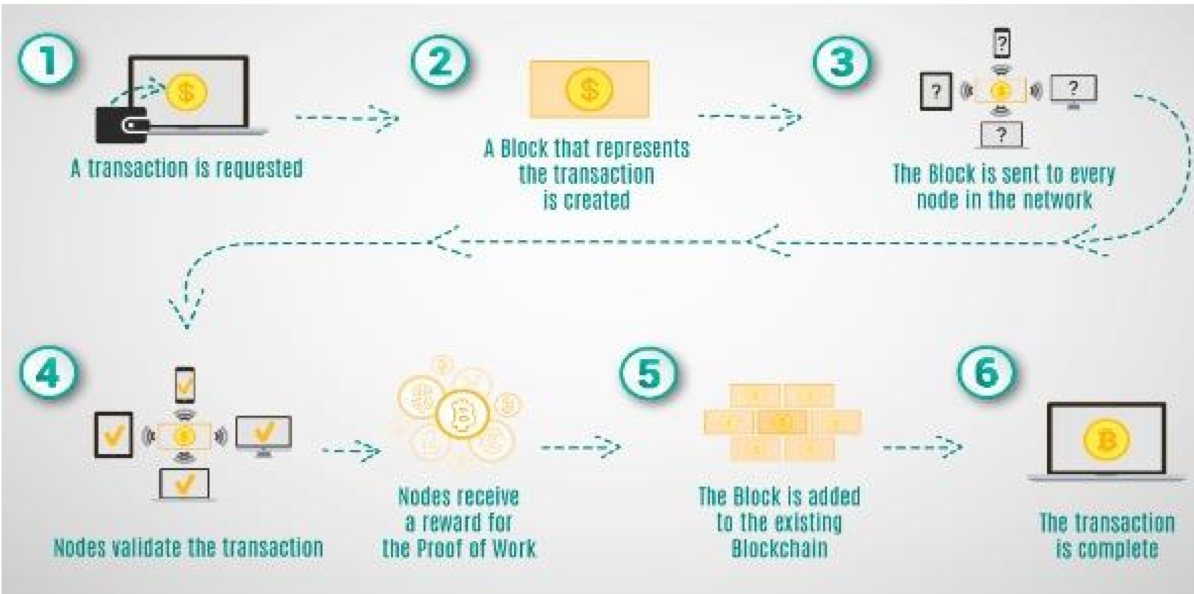
- ❖ Find the best solution to solve existing business problem.
- ❖ Describe the structure,characteristics,behaviour and other aspect of the software to project skectchholder.
- ❖ Define feature,Development phase and Solution requirement.

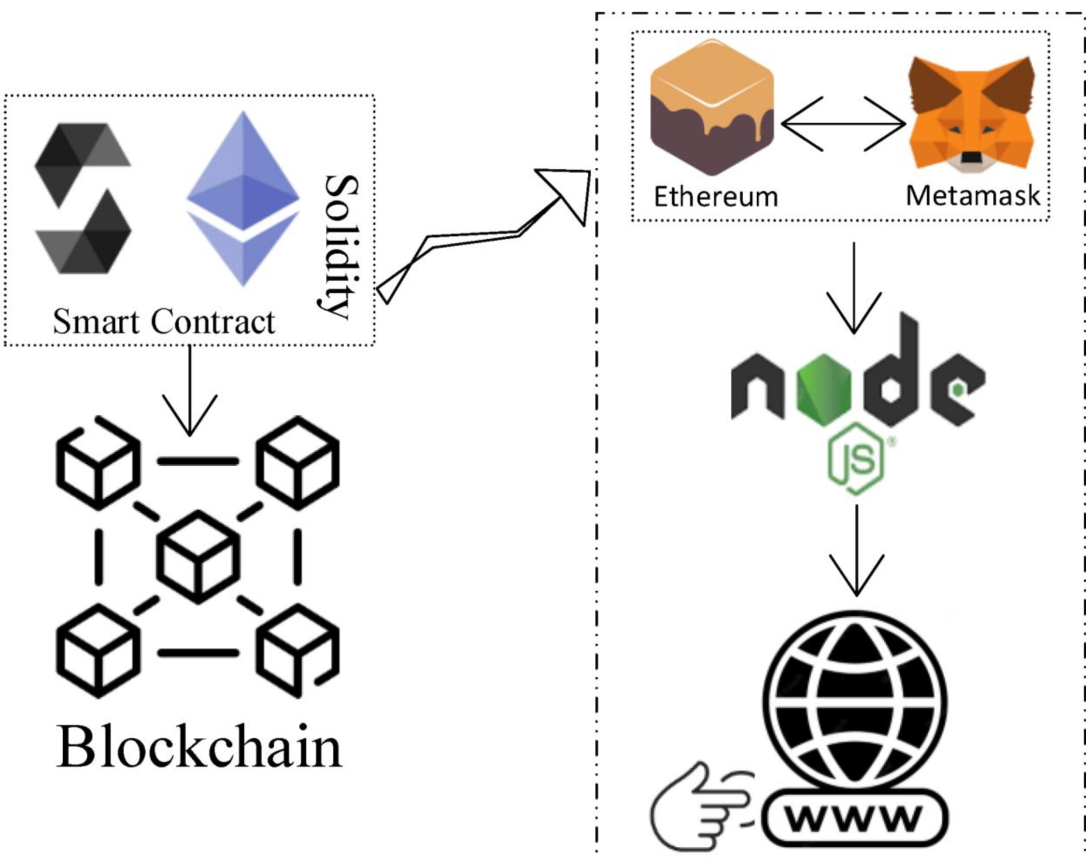
Provide specification according to which the Solution is defined,managed and delivery.



# 6.PROJECT PLANNING & SCHEDULING

## 6.1.Technical Architecture



### 6.2.Sprint Planning & Estimation

Project Initiation and Requirements Gathering-

Tasks: Define project objectives and scope. Identify stakeholders and their requirements.Create a high-level architecture and design.Estimation:Duration: 2 weeks

User Authentication and Registration

Tasks: Implement user registration and authentication. Set up user profiles.Develop a user-friendly interface for user registration.Estimation:Duration: 3 weeks

Blockchain Integration and Data Storage

Tasks:Implement blockchain integration for secure data storage.Develop data upload and storage features for educational institutions.Estimation: Duration: 4 weeks

Credential Verification and Record Sharing

Tasks: Create a credential verification system for employers and students.Enable record sharing and transfer features.Estimation:Duration: 3 weeks

Security and Compliance

Tasks:Enhance system security with encryption and access controls. Ensure compliance with data protection regulations.Estimation:Duration: 2 weeks

Performance Optimization

Tasks:Optimize system performance for scalability. Conduct performance testing and fine-tuning.Estimation:Duration: 2 weeks

User Acceptance Testing and Feedback

Tasks:Invite users to participate in acceptance testing.Gather feedback and make necessary adjustments.Estimation:Duration: 2 weeks

Documentation and Training

Tasks:Create user documentation and training materials.Estimation:Duration: 1 week

Deployment and Go-Live

Tasks:Prepare for system deployment in a production environment.Estimation:Duration: 2 weeks

Post-Deployment Support and Monitoring

Tasks:Provide ongoing support for users.Monitor system performance and security.Estimation:Duration: Ongoing

### 6.3.Sprint Delivery Schedule

(2 weeks): Project Initiation and Requirements Gathering Define project objectives and scope.Identify stakeholders and their requirements.Create a high-level architecture and design.

(3 weeks): User Authentication and Registration,Implement user registration and authentication. Set up user profiles.

(4 weeks): Blockchain Integration and Data Storage Implement blockchain integration for secure data storage Develop data upload and storage features for educational institutions.

(3 weeks): Credential Verification and Record Sharing Create a credential verification system for employers and students.Enable record sharing and transfer features.

(2 weeks): Security and Compliance Enhance system security with encryption and access controls.Ensure compliance with data protection regulations.

(2 weeks): Performance Optimization Optimize system performance for scalability.Conduct performance testing and fine-tuning.

(2 weeks): User Acceptance Testing and Feedback Invite users to participate in acceptance testing. Gather feedback and make necessary adjustments.

(1 week): Documentation and TrainingCreate user documentation and training materials.

(2 weeks): Deployment and Go-LivePrepare for system deployment in a production environment.

(Ongoing): Post-Deployment Support and Monitoring Provide ongoing support for users.Monitor system performance and security.

## 7.CODING & SOLUTIONING (Explain the features added in the project along with code)

### 7.1.Feature1

### Certificate Revocation

This feature allows the contract owner to revoke a certificate in case of fraud or errors.

In this code, we've added a new function `revokeCertificate` that checks if the certificate exists and whether the sender (contract owner) is the issuer. If these conditions are met, the certificate is revoked by deleting the data. An event `CertificateRevoked` is emitted to record the revocation.

**Solidity Code:**

```
function revokeCertificate(uint256 certificateId) external onlyOwner {
    require(certificateId <= totalCertificates, "Certificate not found");
    Certificate storage cert = certificates[certificateId];

    // Additional checks for revocation, e.g., fraud detection
    require(msg.sender == cert.issuer, "Only the issuer can revoke the
certificate");

    // Revoke the certificate by clearing the data
    delete certificates[certificateId];

    emit CertificateRevoked(certificateId, cert.studentName,
cert.courseName, now, msg.sender);
}
```

### 7.2.Feature02
### Certificate Lookup by Student Name

This feature allows anyone to look up a certificate by providing the student's name.

In this code, we've added a new function `getCertificateByStudentName`, which iterates through the stored certificates and returns the certificate ID when it finds a match based on the provided student name. If no match is found, it returns 0 to indicate that the certificate was not found.

**Solidity Program:**

```
function getCertificateByStudentName(string memory studentName)
external view returns (uint256) {
    for (uint256 i = 1; i <= totalCertificates; i++) {
        if (keccak256(abi.encodePacked(certificates[i].studentName)) ==
keccak256(abi.encodePacked(studentName))) {
            return i;
        }
    }
    return 0; // Not found
}
```

## 7.3.Database Schema (if Applicable)

### 1.On-Chain Ethereum Data Schema:

In Ethereum, data is structured as a series of smart contract states. Each state represents the storage of data within a smart contract. In the "collegeCertificate" smart contract, the data schema can be described as follows:`owner`: An Ethereum address representing the owner of the smart contract.`totalCertificates`: An unsigned integer that keeps track of the total number of certificates issued.certificates`: A mapping of `uint256` (certificate ID) to `Certificate` struct.The `Certificate` struct itself contains the following fields:`studentName`: A string representing the name of the student.`courseName`: A string representing the name of the course.`DateOfGraduation`: A `uint256` representing the date of graduation. `issueDate`: A `uint256` representing the date when the certificate was issued.`issuer`: An Ethereum address representing the entity that issued the certificate.
This data schema reflects the on-chain data structure used in the Ethereum blockchain for storing certificate information.

### 2.Off-Chain Ethereum Data:

Off-chain data in an Ethereum-based application typically refers to data that is not stored directly on the blockchain but is referenced or linked to on-chain data. In the context of this smart contract, off-chain data might include additional details about the student, such as contact information or a transcript. This off-chain data would be stored in a traditional database or IPFS (InterPlanetary File System) and referenced in the smart contract using file hashes or other references.

### 3.Asset Management (if applicable):

The provided smart contract does not directly handle asset management such as digital tokens or fungible/non-fungible assets. If you intend to incorporate asset management into the project, additional smart contracts or standards like ERC-20 (for fungible tokens) or ERC-721 (for non-fungible tokens) would need to be implemented to manage assets on the Ethereum blockchain.

# 8.PERFORMANCE TESTING

## 8.1.Performace Metrics

Throughput:

Metric: Transactions per second (TPS)-Description*: Measure the rate at which the system can process certificate issuance and verification transactions. A high TPS is important to handle a large number of requests efficiently.

Latency:

Metric: Response time -Description*: Measure the time it takes for a transaction (e.g., certificate issuance or verification) to be processed and receive a response. Low latency is critical for user satisfaction.

Scalability: - Metric: Scalability under load -Description*: Test how the system performs as the number of users and transactions increases. Ensure that the system can handle growing demands by adding additional nodes or resources.

Resource Utilization:

Metric: CPU and memory usage -Description*: Monitor the utilization of system resources (CPU and memory) during peak loads. Identify potential bottlenecks and ensure efficient resource allocation.

Availability:

Metric: Uptime and downtime-Description: Measure the system's availability over time. Ensure that it is highly available and robust, with minimal downtime.

Stress Testing

Metric: System failure point-Description*: Test the system's limits to identify the point at which it fails or becomes unstable. This helps determine the system's capacity and resilience under extreme conditions.

Security Testing:

Metric: Successful and unsuccessful attacks-Description: Evaluate the system's resistance to common security attacks such as DDoS attacks, data breaches, and unauthorized access.

Load Testing:

Metric: System performance under expected load-Description: Test the system's performance under expected levels of concurrent users and transactions to ensure it meets operational requirements.

Transaction Confirmation Time:

Metric: Time taken to confirm a transaction on the blockchain-Description: Measure how long it takes for a transaction to be confirmed and added to the blockchain. Short confirmation times are important for efficient certificate issuance and verification.

Blockchain Gas Costs:

Metric: Gas costs per transaction-Description*: Calculate the gas costs associated with each transaction. Minimize gas costs to ensure cost-effective operations.

Error Handling and Recovery:

Metric: Error rates and recovery time-Description: Measure the system's ability to handle errors gracefully and recover from failures without data loss or service interruption.

Capacity Planning:

Metric: Resource scalability-Description: Plan for future capacity needs based on performance metrics and expected growth in users and data.

# Result:

## 9.1.Ouput Screenshot:



## 10. ADVANTAGES & DISADVANTAGES

### ADVANTAGES:

1.Data Security and Integrity: Blockchain technology ensures that student records and academic credentials are securely stored and tamper-proof, reducing the risk of data breaches and fraudulent activity.

2.Transparency:The use of blockchain provides transparency in record-keeping, making it easier for students, educational institutions, employers, and regulators to access and verify academic credentials.

3. Efficiency: The system streamlines the verification process, reducing the time and administrative effort required to validate academic credentials.

4. Empowerment of Students:Students gain control over their academic records, deciding who can access and verify their credentials. This puts them in charge of their own data.

5.Reduction in Fraud:The technology can help combat fraudulent degree mills by providing a trusted and immutable source of degree validation.

6. Cost Savings:By automating and simplifying the verification process, educational institutions and employers can reduce operational costs associated with credential validation.

7. Global Accessibility: Blockchain-based records can be accessed from anywhere, making it easier for students to share their credentials with international institutions and employers.

**DISADVANTAGES:**

1. Complexity: Implementing blockchain technology can be complex and require expertise. Setting up and maintaining the system might be challenging for some educational institutions.

2.Scalability:Scalability can be an issue, especially on public blockchains, as the system needs to accommodate a large number of transactions and users, which can lead to increased costs.

3.Regulatory Compliance:Adhering to data protection regulations and ensuring that the system complies with various regional and international standards can be a significant challenge.

4.Access and Inclusion: Some students might not have access to the technology required to interact with blockchain-based systems, potentially excluding certain demographics.

5. Initial Implementation Costs: The initial setup and integration of blockchain technology may require a significant investment of time and resources.

6. Data Recovery: In the event of data loss, recovery from a blockchain can be challenging, and data may be permanently lost.

7.User Adoption: Users, including educational institutions and employers, may be resistant to change or unfamiliar with blockchain technology, which could hinder adoption.

8. Maintenance: Continuous maintenance, security monitoring, and updates are essential to keep the system running efficiently, which can be resource-intensive.

## 11.CONCLUSION:

the "Transparent Education Data Management Using Blockchain" project holds immense promise for revolutionizing the way educational credentials and student records are stored, accessed, and verified. By leveraging the security and transparency of blockchain technology, the project addresses critical challenges such as data security, verification efficiency, and fraud prevention within the education sector. The empowerment of students to control their own academic records and the potential for global accessibility are significant advantages. However, it's crucial to acknowledge the complexities of implementation, scalability concerns, and the need for strict regulatory compliance. As blockchain technology continues to mature, this project represents a noteworthy step towards a more transparent and secure education data management ecosystem.

## 11. FUTURE SCOPE:

The "Transparent Education Data Management Using Blockchain" project has the potential for significant future developments and enhancements. Some of the future scope areas for this project include:

1. Interoperability: The project can expand to enable interoperability with other educational and professional networks. This would allow for seamless data transfer between various institutions and organizations, facilitating a broader exchange of academic credentials.

2. Integration with Learning Platforms: Integrating the blockchain system with existing learning management platforms can streamline the record-keeping process, making it easier for educators to update records and students to access their academic history.

3.Smart Contracts for Verification:Smart contracts can be used to automate and enhance the verification process further. Employers, educational institutions, and regulators can rely on self-executing smart contracts for instant, secure credential validation.

4.Global Credential Verification Standardization:Collaboration with international bodies and governments to establish a standardized protocol for global credential verification can provide greater credibility and acceptance of blockchain-based academic records worldwide.

5.Blockchain-based Diplomas and Certificates:Instead of merely digitizing traditional paper-based certificates, the project can explore creating blockchain-native diplomas and certificates, which are verifiable and secure by design.

6. User-Friendly Mobile Applications:Developing user-friendly mobile applications for students, educational institutions, and employers can promote widespread adoption and ease of use.

7. Extended Ecosystem: The project can extend its ecosystem to cater to other education-related services, such as scholarship and financial aid management, and academic transcripts.

8. Privacy Enhancements: To address privacy concerns, the system can explore more advanced privacy-preserving technologies, allowing individuals to share selected portions of their records without revealing the entire document.

9.Data Recovery Mechanisms: The development of efficient data recovery mechanisms in case of accidental data loss can provide a safety net for the permanent storage of academic records on the blockchain.

10.Blockchain Scalability Solutions: As blockchain scalability is a concern, the project can stay up to date with blockchain advancements and implement scaling solutions as they evolve.

## 12. APPENDIX Source Code

### SOLIDITY CODE:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract collegeCertificate {
    address public owner;

    struct Certificate {
        string studentName;
        string courseName;
        uint256 DateOfGraduation;
        uint256 issueDate;
        address issuer;
    }

    uint256 public totalCertificates;
    mapping(uint256 => Certificate) public certificates;

    event CertificateIssued(
        uint256 indexed certificateId,
        string studentName,
        string courseName,
        uint256 issueDate,
        address indexed issuer
    );

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only contract owner can call
this");
        _;
    }
```

```
      function issueCertificate(
        string memory studentName,
        string memory courseName,
        uint256 _dateOfGraduation,
        uint256 issueDate
    ) external onlyOwner {
        uint256 certificateId = totalCertificates + 1;

        certificates[certificateId] = Certificate({
            studentName: studentName,
            courseName: courseName,
            DateOfGraduation : _dateOfGraduation,
            issueDate: issueDate,
            issuer: msg.sender
        });

        totalCertificates = certificateId;

        emit CertificateIssued(
            certificateId,
            studentName,
            courseName,
            issueDate,
            msg.sender
        );
    }

    function getCertificate(
        uint256 certificateId
    ) external view returns (string memory, string memory, uint256,
uint256, address) {
        Certificate memory cert = certificates[certificateId];
        return (cert.studentName, cert.courseName,
cert.DateOfGraduation, cert.issueDate, cert.issuer);
    }
}
```

**Java code:**

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

const abi = [
 {
 "inputs": [],
 "stateMutability": "nonpayable",
 "type": "constructor"
 },
 {
 "anonymous": false,
 "inputs": [
  {
  "indexed": true,
  "internalType": "uint256",
  "name": "certificateId",
  "type": "uint256"
  },
  {
  "indexed": false,
  "internalType": "string",
  "name": "studentName",
  "type": "string"
  },
  {
  "indexed": false,
  "internalType": "string",
  "name": "courseName",
  "type": "string"
  },
  {
  "indexed": false,
  "internalType": "uint256",
  "name": "issueDate",
  "type": "uint256"
  },
```

```
    {
    "indexed": true,
    "internalType": "address",
    "name": "issuer",
    "type": "address"
   }
  ],
  "name": "CertificateIssued",
  "type": "event"
 },
 {
  "inputs": [
   {
    "internalType": "string",
    "name": "studentName",
    "type": "string"
   },
   {
    "internalType": "string",
    "name": "courseName",
    "type": "string"
   },
   {
    "internalType": "uint256",
    "name": "_dateOfGraduation",
    "type": "uint256"
   },
   {
    "internalType": "uint256",
    "name": "issueDate",
    "type": "uint256"
   }
  ],
  "name": "issueCertificate",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
 },
```

```
{
"inputs": [
 {
  "internalType": "uint256",
  "name": "",
  "type": "uint256"
 }
],
"name": "certificates",
"outputs": [
 {
  "internalType": "string",
  "name": "studentName",
  "type": "string"
 },
 {
  "internalType": "string",
  "name": "courseName",
  "type": "string"
 },
 {
  "internalType": "uint256",
  "name": "DateOfGraduation",
  "type": "uint256"
 },
 {
  "internalType": "uint256",
  "name": "issueDate",
  "type": "uint256"
 },
```

```
   {
    "internalType": "address",
    "name": "issuer",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [
   {
    "internalType": "uint256",
    "name": "certificateId",
    "type": "uint256"
   }
  ],
  "name": "getCertificate",
  "outputs": [
   {
    "internalType": "string",
    "name": "",
    "type": "string"
   },
   {
    "internalType": "string",
    "name": "",
    "type": "string"
   },
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   },
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   },
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ]
```

```
 {
  "inputs": [],
  "name": "owner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "totalCertificates",
  "outputs": [
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   }
  ],
  "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 =
"0x76118a37cCbf2b99Cc371F9E1B5017065103d5c1"

export const contract = new ethers.Contract(address, abi, signer)
```

**HTML CODE:**

```
<!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>
```

**Github:**

https://github.com/rubadevi1033/Transparent-Education-Data-Management

**Project Video Demo Link:**

https://drive.google.com/file/d/1MW_13Tgq_gPxsKFrUFAxt2Db6C7FPDrr/view?usp=sharing