ANNA UNIVERSITY

NAAN MUDHALVAN – GUIDED PROJECT

TRANSPARENT EDUCATION DATA MANAGEMENT DOCUMENTATION

DATE	31 OCTOBER 2023
TEAM MEMBERS WITH NM ID	1) DHARUN M - 0270CBCD3358D93874A40B917EEC65EC 2) DIVYA DHARSHINI – 087EE4C0342F965ADBA46 49B908C2642 3) ELASUKUMARAN E - 23556780714CDCFB5B84BF4E9269022A 4) GOWTHAM R – 0BF3F06AC53C10E89F446EF2C7C77B3C 5) VYSHNAVI M – 4284E4FD1316A93BEA54DC B745B8F87F
PROJECT NAME	TRANSPARENT EDUCATION DATA MANAGEMENT
COLLEGE NAME	ALAGAPPA COLLEGE OF TECHNOLOGY, ANNA UNIVERSITY

1. INTRODUCTION:

BLOCKCHAIN:

Blockchain is like a digital ledger that records transactions across multiple computers in a secure and transparent way. It's the tech behind cryptocurrencies, but its applications go beyond that to areas like smart contracts and decentralized apps.

1.1 Project Overview:

Transparent Education Data Management is a comprehensive initiative aimed at improving the transparency and efficiency of educational data management systems. This project addresses the critical need for accessible, accurate, and secure data in the education sector, enabling educational institutions, policymakers, and stakeholders to make informed decisions and enhance the overall quality of education.

1.2 Purpose:

The project aims to increase transparency within the education sector by providing stakeholders with real-time access to accurate and up-to-date educational data. This transparency helps ensure that the data remains open and accessible to those who need it. Transparent Education Data Management. It seeks to minimize errors and discrepancies in educational data, leading to more reliable and trustworthy information for decision-making. It empowers educational institutions, policymakers, and other stakeholders to make informed, data-driven decisions to enhance the quality of education. Ultimately, the project's purpose is to contribute to the improvement of educational outcomes by providing the tools and information necessary for institutions and policymakers to allocate resources effectively and make informed decisions.

2. LITERATURE SURVEY:

2.1 Existing Problems:

Educational data is often scattered across various departments and systems within educational institutions, making it challenging to access and consolidate for comprehensive analysis. Manual data entry and outdated systems lead to inaccuracies in educational data, which can result in incorrect decision-making and resource allocation. With an increasing amount of sensitive student and institutional data being stored digitally, concerns about data security, privacy, and compliance with data protection regulations are significant issues. Different departments within educational institutions often maintain their own isolated data systems, which makes it difficult to share data and gain a holistic view of educational performance. Many educational systems rely on outdated or batch-based reporting, resulting in delayed access to critical information needed for timely decision-making. Stakeholders, including students, parents, and policymakers, often lack access to detailed educational data, leading to transparency gaps and a lack of trust in the system.

2.2 Reference:

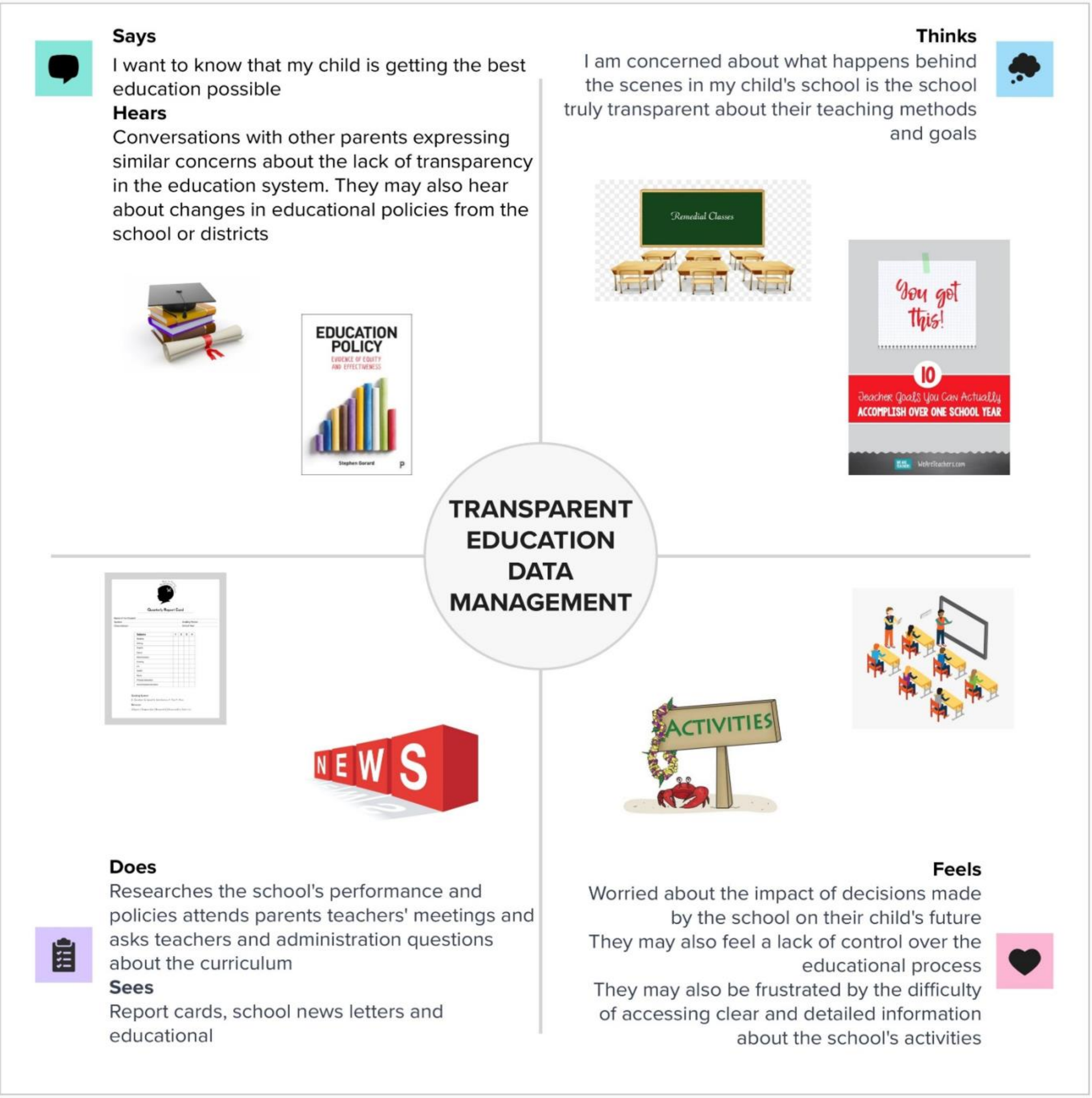
S. No.	Literature	Author
1	Transparency in Education: An International Study of Accountability in Higher Education	Stephen P. Heyneman and Kathryn F. Connolly.
2	Measuring What Matters in Education	Organization for Economic Co-operation and Development (OECD).
3	Data Management for Researchers: Organize, maintain and share your data for research success	Kristin Briney.
4	Data Analytics for Education	Ryan S.J.D. Baker and George Siemens.

2.3 Problem Statement Definition:

The current state of education data management faces a multitude of challenges that impede the effective functioning of educational institutions and hinder data-driven decision-making. These challenges include data fragmentation, inaccuracy, security concerns, limited accessibility, data silos, lack of real-time data, integration issues, resource allocation inefficiencies, transparency gaps, regulatory compliance hurdles, data redundancy, and limited data analytics capabilities. The problem at hand is the need to develop and implement a comprehensive and innovative Transparent Education Data Management system that addresses these issues. This system should ensure the accessibility, accuracy, security, and transparency of educational data while promoting data-driven improvements in the education sector. The lack of such a system hampers educational institutions, educators, students, and policymakers in their efforts to make informed decisions, allocate resources efficiently, and build trust in the education system. The project aims to provide a solution to these pressing challenges and transform the management of educational data for the betterment of the entire education sector.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



Says

I want to know that my child is getting the best education possible

Hears

Conversations with other parents expressing similar concerns about the lack of transparency in the education system. They may also hear about changes in educational policies from the school or districts

Thinks

I am concerned about what happens behind the scenes in my child's school is the school truly transparent about their teaching methods and goals

TRANSPARENT EDUCATION DATA MANAGEMENT

Does

Researches the school's performance and policies attends parents teachers' meetings and asks teachers and administration questions about the curriculum

Sees

Report cards, school news letters and educational

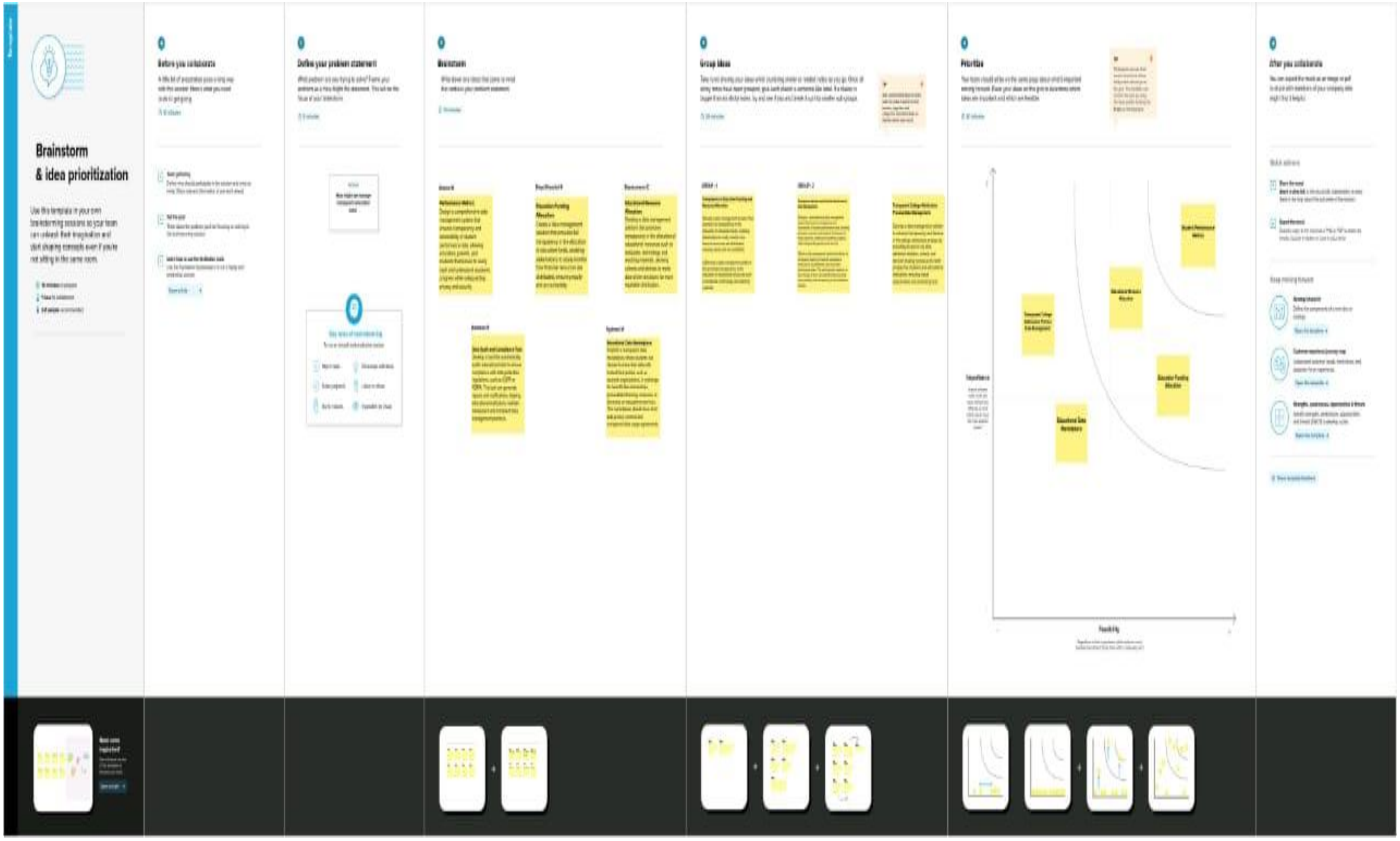
Feels

Worried about the impact of decisions made by the school on their child's future

They may also feel a lack of control over the educational process

They may also be frustrated by the difficulty of accessing clear and detailed information about the school's activities

3.2 Ideation & Brainstorming:



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.

Before you collaborate

Define your problem statement

Key rules of brainstorming

Brainstorm

Performance Metrics

Education Funding Allocation

Educational Resource Allocation

Data Audit and Compliance Tool

Educational Data Marketplace

Group ideas

Transparent College Admission Process Data Management

Prioritize

Student Performance Metrics

Educational Resource Allocation

Transparent College Admission Process Data Management

Education Funding Allocation

Educational Data Marketplace

Importance

Feasibility

After you collaborate

4. REQUIREMENT ANALYSIS:

4.1 Functional requirements:

The functional requirements of this project are Visual Studio Code, one remix id platform (node.js connector), file explorer, meta mask chrome extension and a source code file.

When designing a blockchain-based project for "Transparent Education Data Management," you need to define both functional and non-functional requirements to ensure that the system effectively meets its objectives. Here are requirements for such a project:

4.1 Functional Requirements:

1. **User Registration and Authentication:**

- Implement a user registration system for students, teachers, administrators, and other stakeholders.

- Enable secure user authentication using cryptographic techniques, such as public and private keys.

2. **Data Storage and Immutability**:

- Use blockchain technology to store educational data in a decentralized and immutable ledger.

- Ensure that once data is added to the blockchain, it cannot be altered, ensuring data integrity.

3. **Data Ingestion and Verification:**

- Develop a mechanism for ingesting data from educational institutions and verifying its authenticity.

- Implement data validation and verification processes to ensure the accuracy and integrity of the data.

4. **User Access Control:**

- Define access control mechanisms using smart contracts to specify who can access, modify, and view specific data.

- Ensure that roles and permissions are programmatically enforced.

5. **Data Retrieval and Transparency:**

- Enable users to retrieve and view educational data with appropriate permissions.

- Create a transparent system where stakeholders can trace data changes and access a complete audit trail.

6. **Reporting and Analytics:**

- Implement reporting and analytical tools to provide insights into student performance, attendance, and other educational metrics.

- Leverage smart contracts and data analysis to generate real-time and historical reports.

7. **Communication and Notifications:**

- Develop messaging and notification features to support communication among students, teachers, administrators, and parents.

8. **Security:**

- Use blockchain's inherent security features, such as cryptographic hashing and decentralized consensus, to protect data from unauthorized access or tampering.

9. **Integration with External Systems:**

- Create APIs or interfaces for integration with other educational systems and databases.

- Ensure data interoperability and seamless exchange of information.

10. **Data Privacy and Compliance:**

- Implement encryption and hashing techniques to safeguard sensitive data, adhering to relevant data protection regulations, such as GDPR or FERPA.

11. **Smart Contracts**:

- Develop and deploy smart contracts for executing specific actions, such as data validation, access control, and automated processes.

12. **Scalability and Performance:**

- Design the blockchain infrastructure to scale as the system grows, accommodating a potentially large volume of educational data.

Drive link of the source code: https://drive.google.Com/file/d/15VKMwX8Ff-XgETI2v2tlkpNXS6Aw5Aeb/view?Usp=sharing

4.2 Non-Functional requirements:

Non-functional requirements (NFRs) are essential for ensuring that the project meets its performance, security, and usability goals.The project should have an intuitive and user-friendly interface to ensure ease of use. It should be highly available and reliable, with minimal downtime for maintenance or updates. It should be scalable to handle increased data and user loads over time. . Here are some non-functional requirements that could be relevant to the project:

1. Performance:

- **Response Time:** The system should provide quick response times for queries and data retrieval, ensuring that users can access information promptly.

- **Scalability:** The system should be able to handle increased loads as the user base and data volume grow.

- **Availability:** The system should be available 24/7, with minimal downtime for maintenance or updates**.**

- **Throughput**: The system should handle a certain number of concurrent users and transactions without performance degradation.

2. Security:

- **Data Encryption**: All sensitive data, including student and staff information, should be encrypted both in transit and at rest.

- **Authentication and Authorization:** The system should enforce robust user authentication and authorization mechanisms to ensure that only authorized personnel can access specific data.

- **Data Privacy:** Compliance with data protection regulations and standards, such as GDPR or HIPAA, should be ensured where applicable**.**

- **Audit Trails:** The system should maintain audit logs to track data access and changes for accountability and security purposes**.**

3. Reliability:

- **Fault Tolerance:** The system should continue to function in the presence of hardware or software failures without losing data or compromising the user experience.

- **Backup and Recovery:** Regular automated backups and a well-defined recovery process should be in place to mitigate data loss in case of unexpected events.

4. Scalability:

- **Horizontal Scalability:** The system should be designed to scale horizontally by adding more servers or resources to handle increased demand.

- **Vertical Scalability:** The system should also be able to scale vertically by utilizing more powerful hardware if required**.**

5. Usability:

- **User Interface (UI) Design:**The user interface should be intuitive and user-friendly, promoting ease of use for both technical and non-technical users**.**

- **Accessibility:**The system should comply with accessibility standards to ensure it can be used by individuals with disabilities**.**

6. Interoperability:

- Integration: The system should be able to integrate with other educational software, systems, or databases to exchange data efficiently**.**

- Data Exchange Formats: Support for standard data exchange formats (e.g., XML, JSON) should be provided for interoperability with external systems.

7. Compliance:

- Regulatory Compliance: Ensure that the system complies with relevant laws and regulations related to education data management, such as FERPA (Family Educational Rights and Privacy Act).

8. Performance Testing:

- Load Testing: Conduct load testing to ensure the system can handle the expected number of users and data volume**.**

- Stress Testing: Assess how the system performs under extreme conditions to identify potential bottlenecks or failures.

9. Documentation:

- User Documentation: Provide comprehensive user documentation and training materials to assist users in navigating and using the system effectively.

- System Documentation: Maintain detailed technical documentation for system administrators and support personnel**.**

10. Monitoring and Reporting:

- Implement monitoring tools and generate reports for system performance, usage patterns, and potential issues, enabling proactive management and decision-making.

5. PROJECT DESIGN:

5.1 Data Flow Diagram & User Stories

Characteristic of blockchain technologies:

Some key characteristics of blockchain technologies are listed as follows:

→ **Increased capacity**: Blockchain technology can increase the capacity of an entire network. One such example is the supercomputer created by Stanford University usedfor medical research.

→ **Better security:** Blockchain technology offers better security as it provides for a network of numerous computer nodes that can be used for networking transactions .

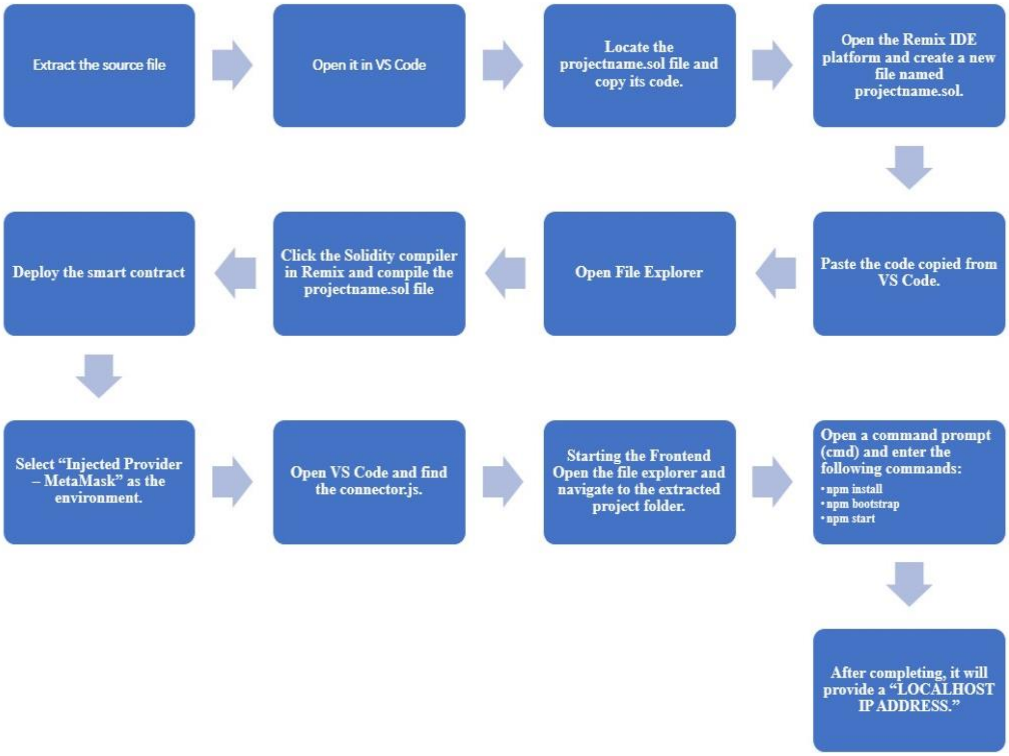
→ **Immutability:** Blockchain uses immutable ledgers, and all databases require trust of a third party to keep them secure from hackers. Blockchain applications, such as Bitcoin, maintain the ledger in a never-ending state of forward momentum.

→ **Faster settlement**: Blockchain technology relies on faster speeds and saves time for institutions and consumers. One example from banking is that blockchain makes money transfer fast and convenient.

→ **Decentralized System:** Blockchain technology offers a decentralized system that stores the assets in a network and can be accessed via the internet. The asset may be a contract or document of importance. The manager of blockchain technology has control over the accounts of individuals and can transfer anything to anyone. This technology is proving to be an effective tool for decentralizing the web.

→ **Minting:** Blockchain technology involves minting a problem in several ways. Proof of work is one approach guaranteeing an individual is engaged in a significant amount of computation work.

5.1.1 Data Flow Diagram


Extract the source file
Open it in VS Code
Locate the projectname.sol file and copy its code.
Open the Remix IDE platform and create a new file named projectname.sol.
Paste the code copied from VS Code.
Open File Explorer
Click the Solidity compiler in Remix and compile the projectname.sol file
Deploy the smart contract
Select "Injected Provider – MetaMask" as the environment.
Open VS Code and find the connector.js.
Starting the Frontend Open the file explorer and navigate to the extracted project folder.
Open a command prompt (cmd) and enter the following commands:
• npm install
• npm bootstrap
• npm start
After completing, it will provide a "LOCALHOST IP ADDRESS."

Source Data flow diagram

5.1.2 User Stories:

User stories are a useful way to outline the functional requirements of your project from the perspective of end-users and stakeholders.

- ✓ As an Administrator, I want user Account Management, create and manage user accounts, assign roles, and reset passwords as needed. Disable or delete user accounts when necessary.

- ✓ As an Educator, I want data entry and management, enter student grades, attendance records, and assessment results into the system. View and edit student information and performance data.
- ✓ As a Student, I want to access to Personal Data, view my own grades, attendance records, and assessment results. Access any relevant announcements or assignments from educators.
- ✓ As a Parent, I want to access to Child's Data, view my child's grades, attendance records, and assessment results. Receive notifications and updates on my child's performance.

5.2 Solution Architecture:

Implement a centralized database or data warehouse that serves as the primary repository for all educational data, including student records, assessment results, attendance, and institutional performance data. This centralization ensures data consistency and accessibility. Employ robust user authentication and authorization mechanisms to control access to the system. Implement role-based access control, allowing administrators, educators, students, and parents to access specific data and functionalities based on their roles. Develop data integration processes and ETL workflows to consolidate data from various sources. These processes should include data mapping, transformation, and validation to ensure data accuracy before it is stored in the central repository. Build a data analytics and reporting engine that allows users to create, customize, and schedule reports and visualizations. Implement data analytics tools to derive meaningful insights from the educational data, aiding data-driven decision-making.

Prerequisite:

1. download node.js : Node.js
2. download vs code: Li4nk
3. download metamask : https://metamask.io/

Steps to complete the project

Step 1:-

1. Open the Zip file and download the zip file.

2. Extract all zip files

Step 2 :

1.Open vs code in the left top select open folder. Select extracted file and open .

2. Select the projectname.sol file and copy the code.

3. Open the remix ide platform and create a new file by giving the name of projectname.sol and paste the code which you copied from vs code.

4. Click on solidity compiler and click compile the projectname.sol

5. Deploy the smart contract by clicking on the deploy and run transaction.

6. select injected provider - MetaMask. In environment

7. Click on deploy. Automatically MetaMask will open and give confirmation. You will get a pop up click on ok.

8. In the Deployed contract you can see one address copy the address.

9. Open vs code and search for the connector.js. In contract.js you can paste the address at the bottom of the code. In export const address.

10. Save the code.

Step 3: open file explorer

1. Open the extracted file and click on the folder.

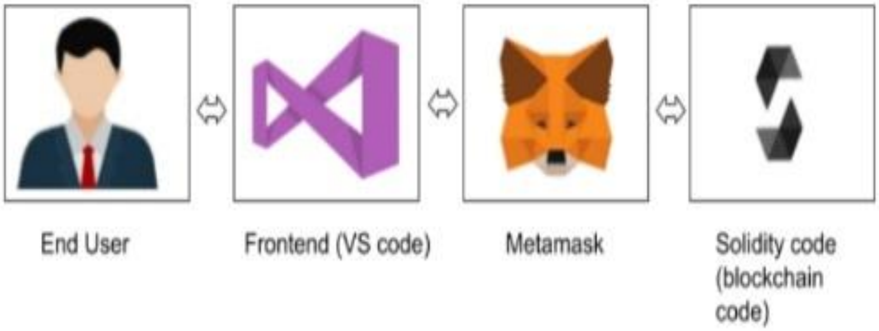
2. Open src, and search for utiles.

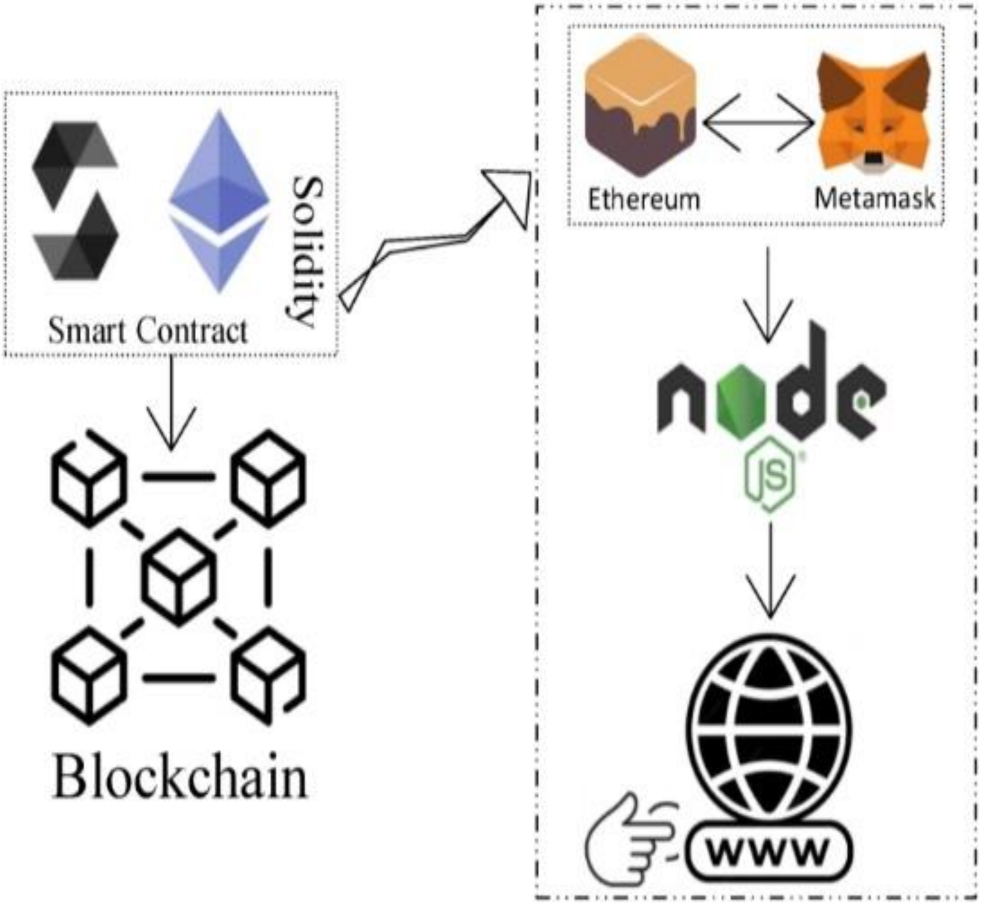
3 . You can see the frontend files. Select all the things at the top in the search bar by clicking alt+ A. Search for cmd

4. Open cmd enter commands

```
npm install

npm bootstrap

npm start
```

5. It will install all the packages and after completing it will open {LOCALHOST IP ADDRESS} copy the address and open it to chrome so you can see the frontend of your project.





Schematic Diagram of Solution Architecture

6. PROJECT PLANNING & SCHEDULING:

6.1 Technical Architecture:

Develop a responsive web-based frontend using technologies like HTML, CSS, and JavaScript. Implement a user-friendly interface that allows administrators, educators, students, and parents to access the system through web browsers. Create a backend server using a framework like Node.js, Django, or Ruby on Rails. This server will handle user authentication, data processing, data integration, validation, and communication with the database. Implement a relational database or data warehouse to store and manage educational data. Use database management systems like PostgreSQL, MySQL, or Microsoft SQL Server. Ensure that the database schema supports data integration and is optimized for efficient data retrieval. Integrate data analytics and reporting tools, such as Tableau, Power BI, or custom-built solutions, into the architecture. These tools should allow users to create, customize, and schedule reports and visualizations based on the educational data stored in the database. Host the application and database on a secure and scalable cloud platform, such as AWS, Azure, or Google Cloud. Implement security measures, including encryption, role-based access control, and firewall configurations. Regularly update and maintain the infrastructure to ensure high availability and reliability.

Project Overview: The project aims to develop a secure and transparent blockchain-based voting system. It includes technical architecture design, sprint planning, estimation, and a sprint delivery schedule.

Objective: Design the technical architecture for the blockchain voting system.

Tasks:

Project Kickoff (Week 1):

- Define project scope, objectives, and key stakeholders.
- Assemble project team.

Requirements Analysis(Week 2-3):

- Gather functional and non-functional requirements.
- Define the scope of blockchain integration.

Technical Design(Week 4-5):

- Create a high-level system architecture.
- Select the blockchain platform (e.g.,Ethereum, Hyperledger) and consensus mechanism.
- Design data models and smart contracts.
- Define security protocols and privacy measures.

System Prototyping(Week 6-8):

- Develop a prototype to validate the technical design.
- Conduct initial security and performance tests.

Technical Architecture Review(Week 9):

- Review the technical architecture with stakeholders.

Address feedback and finalize the design

6.2 Sprint Planning & Estimation:

Before each sprint, conduct backlog refinement to review and prioritize user stories. Work with stakeholders to identify and prioritize the most critical features and improvements based on their importance and impact. Ensure that the sprint goal is specific and measurable, making it easier to track progress. Use a relative estimation technique, such as story points or ideal days, to estimate the effort required for each user story. Involve the development team in the estimation process to gain a consensus on the effort require in a sprint planning meeting, select a set of user stories from the prioritized backlog that can be realistically completed in the upcoming sprint. Break down user stories into tasks and define acceptance criteria for each. Used velocity to determine the number of story points or tasks that can be taken into the sprint based on the sprint duration.

6.3 Sprint Delivery Schedule:

Sprint 1 (Duration: 1 week)

Sprint Goal:

Set up the foundational architecture for the system.

User Stories:

User authentication and role-based access control.
Database schema design for central data repository.
Basic user interface for administrators.

User Story Estimations:

12 story points.

Deliverables:

Authentication system, basic database structure, and administrator login functionality.

Sprint 2 (Duration: 1 week)

Sprint Goal:

Implement data integration and validation processes.

User Stories:

Data integration from one data source (e.g., student records).
Data validation and error handling.

User Story Estimations:

10 story points.

Deliverables:

Data integration module for one data source, validation framework.

Sprint 3 (Duration: 1 week)

Sprint Goal:

Enhance data management and analytics.

User Stories:

Complete data integration for additional data sources (e.g., attendance, assessments).

Implement basic reporting and visualization features.

User Story Estimations:

15 story points.

Deliverables:

Data integration for additional sources, basic reporting tools.

Sprint 4 (Duration: 1 week)

Sprint Goal:

Improve data security and user management.

User Stories:

Implement data encryption and access controls.

User account management features (create, reset password, etc.).

User Story Estimations:

14 story points.

Deliverables:

Enhanced data security and user management capabilities.

Sprint 5 (Duration: 1 week)

Sprint Goal:

Enhance the user interface and user experience.

User Stories:

Improve the user interface for educators, students, and parents.

Implement user notifications and communication features.

User Story Estimations:

10 story points.

Deliverables:

Improved user interfaces and communication features.

7. CODING AND SOLUTIONING:

7.1 Feature:

Transparent education data management

Description:

In the context of education data management, this feature allows administrators, staff members, students and as well as parents

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);
    }
}
```

7.2 Database Schema:

The database schema for the Transparent Education Data Management project is a crucial aspect of the database design. It defines the structure of the database and how data is organized within it.

Students Table:

This table stores information about individual students.
Fields may include:

- **student_id**
- **first_name**
- **last_name**
- **date_of_birth**
- **gender**
- **contact_email**
- **contact_phone**

Courses Table:

This table holds details about the courses offered in the educational institution.
Fields may include:

- **course_id**
- **course_name**
- **instructor_id**
- **start_date**
- **end_date**

Enrollments Table:

This table represents the enrollment of students in courses.
Fields may include:

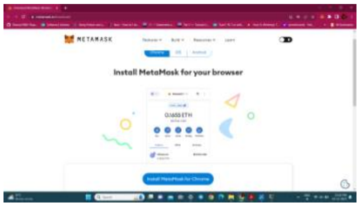
- **enrollment_id**
- **student_id**
- **course_id**
- **enrollment_date**

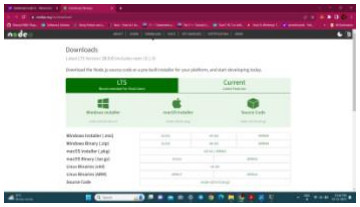
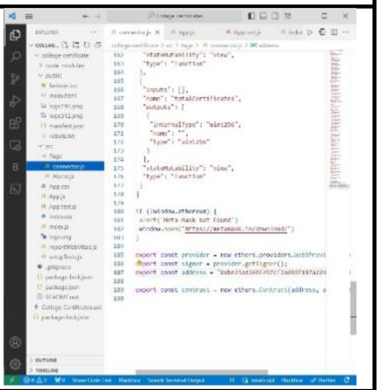
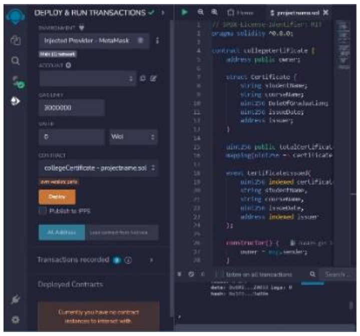
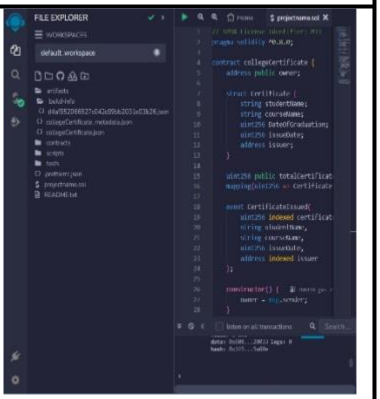
Grades Table:
This table records students' grades for assessments within courses.
Fields may include:

- **grade_id**
- **enrollment_id**
- **assessment_name**
- **score**
- **assessment_date**

8. PERFOMANCE TESTING:

8.1 Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite:	Meta Mask Installed  Install MetaMask for your browser VS Code Installed Download Visual Studio Code Free and built on open source. Integrated Git, debugging and extensions. Visual Studio Code Download Windows .deb .rpm Mac

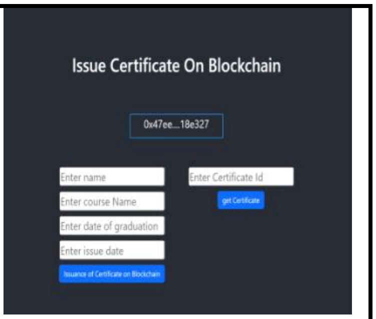
			Node js Installed 
2.	Extract the zip files	Open to VS code	
3.	Remix IDE Platform exploring	Deploy the smart contract code Deploy and run the transaction. By selecting the environment - inject the MetaMask.	
4	Open file explorer	Open the extracted file and click on the folder Open src, and search for utiles. Open cmd enter command 1. npm install 2. npm bootstrap 3. npm start	

```
    "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.Web3Provi
export const signer = provider.getSigner();
export const address = "0xbe21a6d88f407cf2a0807187a220
export const contract = new ethers.Contract(address, a
```

DEPLOY & RUN TRANSACTIONS
Injected Provider - MetaMask
3000000
Wei
collegeCertificate - projectname.sol
Deploy
Publish to IPFS
At Address
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.
projectname.sol

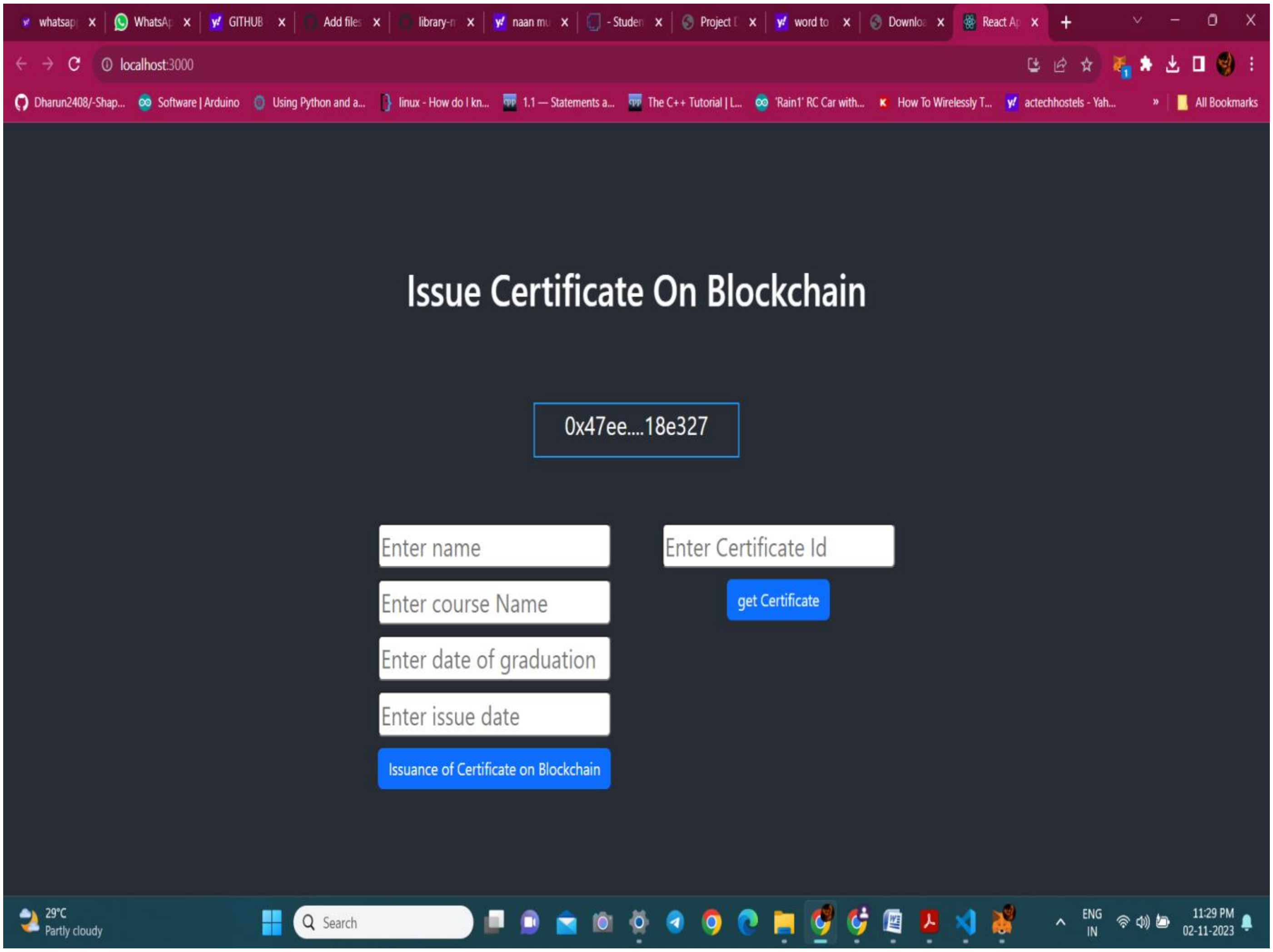
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract collegeCertificate {
    address public owner;

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

    uint256 public totalCertificat
    mapping(uint256 => Certificate

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

    constructor() {
        owner = msg.sender;
    }
```

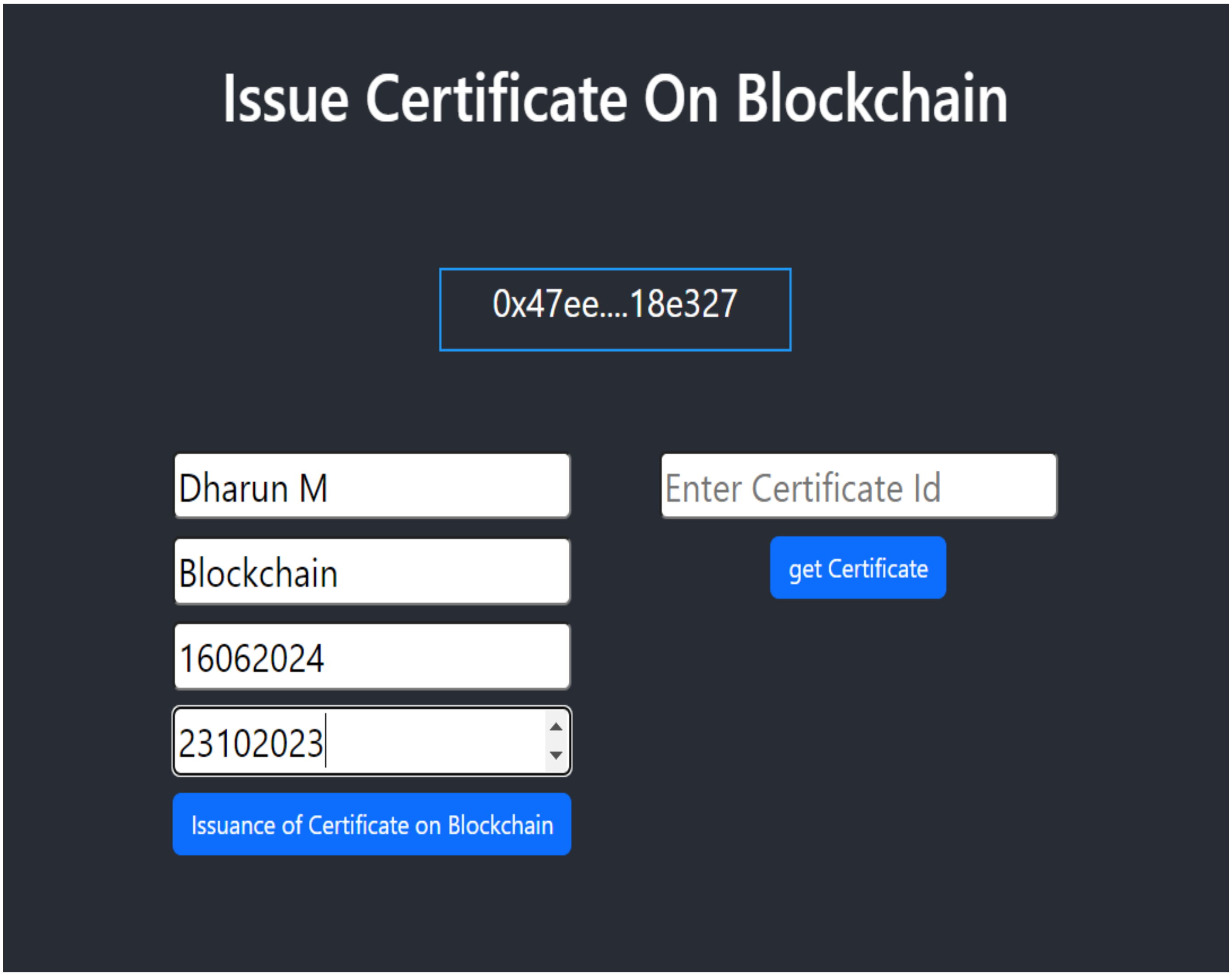
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract collegeCertificate {
    address public owner;

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

    uint256 public totalCertificat
    mapping(uint256 => Certificate

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

    constructor() {
        owner = msg.sender;
    }
```

5	LOCAL HOST IP ADDRESS	Copy the address and open it to chrome so you can see the front end of your project.	 Issue Certificate On Blockchain 0x47ee....18e327 Enter name Enter course Name Enter date of graduation Enter issue date Issuance of Certificate on Blockchain Enter Certificate Id get Certificate

9. RESULT:

9.1 Output Screenshot





Issue Certificate On Blockchain

0x47ee....18e327

Dharun M

Blockchain

16062024

23102023

10. ADVANTAGES AND DISADVANTGES:

10.1 Advantages:

- The project enables educational institutions to make data-driven decisions, leading to better resource allocation and improved educational outcomes.
- The system enhances transparency and accountability, building trust between educational institutions, educators, students, and parents.
- It streamlines data management, reducing data redundancy and improving data accuracy through automation and validation.
- Users can create customized reports and visualizations, gaining valuable insights into student performance and educational trends.
- The project incorporates robust data security measures, ensuring compliance with data protection regulations and safeguarding sensitive information.

10.2 Disadvantages:

- Implementing the system can be complex and time-consuming, requiring significant resources and expertise.
- Managing sensitive student data raises privacy concerns that need to be addressed effectively to prevent data breaches.
- The project may involve significant initial costs for software development, hardware, and infrastructure.
- Training users and administrators to effectively utilize the system may take time and resources.
- Ongoing maintenance and regular updates are necessary to keep the system secure and operational, potentially incurring additional costs.

11. CONCLUSION:

The Transparent Education Data Management project represents a significant step toward modernizing and enhancing the educational sector. By addressing the challenges of data fragmentation, inaccuracy, security concerns, and limited accessibility, the project strives to empower educational institutions, educators, students, and parents with a robust system for data-driven decision-making.

The advantages of this project, such as data-driven decision-making, transparency, efficient data management, customized reporting, and data security, underscore its potential to improve the quality of education and foster greater trust and accountability within the system.

However, it is important to acknowledge the potential disadvantages, including implementation challenges, data privacy concerns, initial costs, user training requirements, and ongoing maintenance needs. These challenges highlight the need for careful planning, security measures, and the allocation of appropriate resources to ensure the project's success.

In conclusion, the Transparent Education Data Management project has the potential to revolutionize education data management, providing a platform that fosters transparency, efficiency, and informed decision-making. Its successful implementation and operation will depend on the dedication, expertise, and resources invested in addressing both the advantages and disadvantages, ultimately contributing to the advancement of education and the well-being of students and educational institutions.

12. FUTURE SCOPE:

- The future scope of the Transparent Education Data Management project is promising, with numerous opportunities for growth and enhancement.
- Integrating the system with popular Learning Management Systems (e.g., Moodle, Canvas, Blackboard) can provide a seamless experience for educators and students, allowing for a comprehensive view of educational data.
- Develop a mobile application to make educational data accessible on mobile devices, allowing students and parents to track progress and receive notifications on the go.
- Enhance communication features for educators, students, and parents, enabling real-time feedback, alerts, and collaboration. Utilize artificial intelligence (AI) to provide deeper insights into educational data, such as identifying trends, suggesting improvements, and optimizing resource allocation.
- Explore blockchain technology to enhance data security, ensuring data immutability and traceability while maintaining privacy.

APPENDIX:

SOURCE CODE:

The source code of the project is given in the below drive link.

Source code link: https://drive.google.Com/file/d/15VKMwX8Ff-XgETI2v2tlkpNXS6Aw5Aeb/view?Usp=sharing

GITHUB LINK:

https://github.com/blockchain-nm/Transparent-Education-Data-Management/tree/main

DEMONSTRATION VIDEO LINK:

https://drive.google.com/drive/folders/1y_V_SAstNPfZgB1zGvAr9qPeg-d-b6uw