TRANSPARENT EDUCATION DATA MANAGENENT

DOMAIN – BLOCK CHAIN

PROJECT REPORT

SUBMITTED BY

TEAM ID: NM2023TMID11953

MENTOR : P.Caroline Priyanka Evelyne

ARUL PRAKASH A - 812020106003

MANIYARASAN V - 812020106303

PRAVEEN P - 812020106029

SHUBUHATHDULLAH S - 812020106501

SRIKANTH R - 812020106034

In the partial fulfilment of the requirements for the award of a degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

MAM COLLEGE OF ENGINEERING AND TECHNOLOGY

Project Report Format

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

3. REQUIREMENT ANALYSIS

3.1 Functional requirement

3.2 Non-Functional requirements

4. PROJECT DESIGN

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

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

5.1 Feature 1

5.2 Feature 2

5.3 Database Schema (if Applicable)

6. RESULTS

6.1 Performance Metrics

7. ADVANTAGES & DISADVANTAGES

8. CONCLUSION

9. FUTURE SCOPE

10. APPENDIX

Source Code
GitHub & Project Video Demo Link

1.INTRODUCTION

A blockchain is “a distributed database that maintains a continuously growing list of ordered records, called blocks.” These blocks “are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network.

1.1. Project Overview

The wealth of user data acts as a fuel for network intelligence toward the sixth generation wireless networks (6G). Due to data heterogeneity and dynamics, decentralized data management (DM) is desirable for achieving transparent data operations across network domains, and blockchain can be a promising solution. However, the increasing data volume and stringent data privacy-preservation requirements in 6G bring significantly technical challenge to balance transparency, efficiency, and privacy requirements in decentralized blockchain-based DM. In this paper, we investigate blockchain solutions to address the challenge. First, we explore the consensus protocols and scalability mechanisms in blockchains and discuss the roles of DM stakeholders in blockchain architectures. Second, we investigate the authentication and authorization requirements for DM stakeholders. Third, we categorize DM privacy requirements and study blockchain-

based mechanisms for collaborative data processing. Subsequently, we present research issues and potential solutions for blockchain-based DM toward 6G from these three perspectives. Finally, we conclude this paper and discuss future research directions.

1.2. Purpose

Blockchain technology can be used in secure and transparent data management by providing a decentralized ledger for recording transactions. This eliminates the need for intermediaries, reducing the risk of data breaches and cyber-attacks. Blockchain technology shows promise for those government bodies that are looking for better ways to manage and protect trusted information. It offers an enticing path toward more efficient operations, more responsive service, and enhanced data security. As early adopters in financial-service industries can attest, however, it will take time for the technology to fully mature. Now is the time for experimentation. By including blockchain in their innovation agendas—establishing it as a critical component of enterprise architecture—governments will learn what works in practice and how to unlock the full potential of data-driven service.

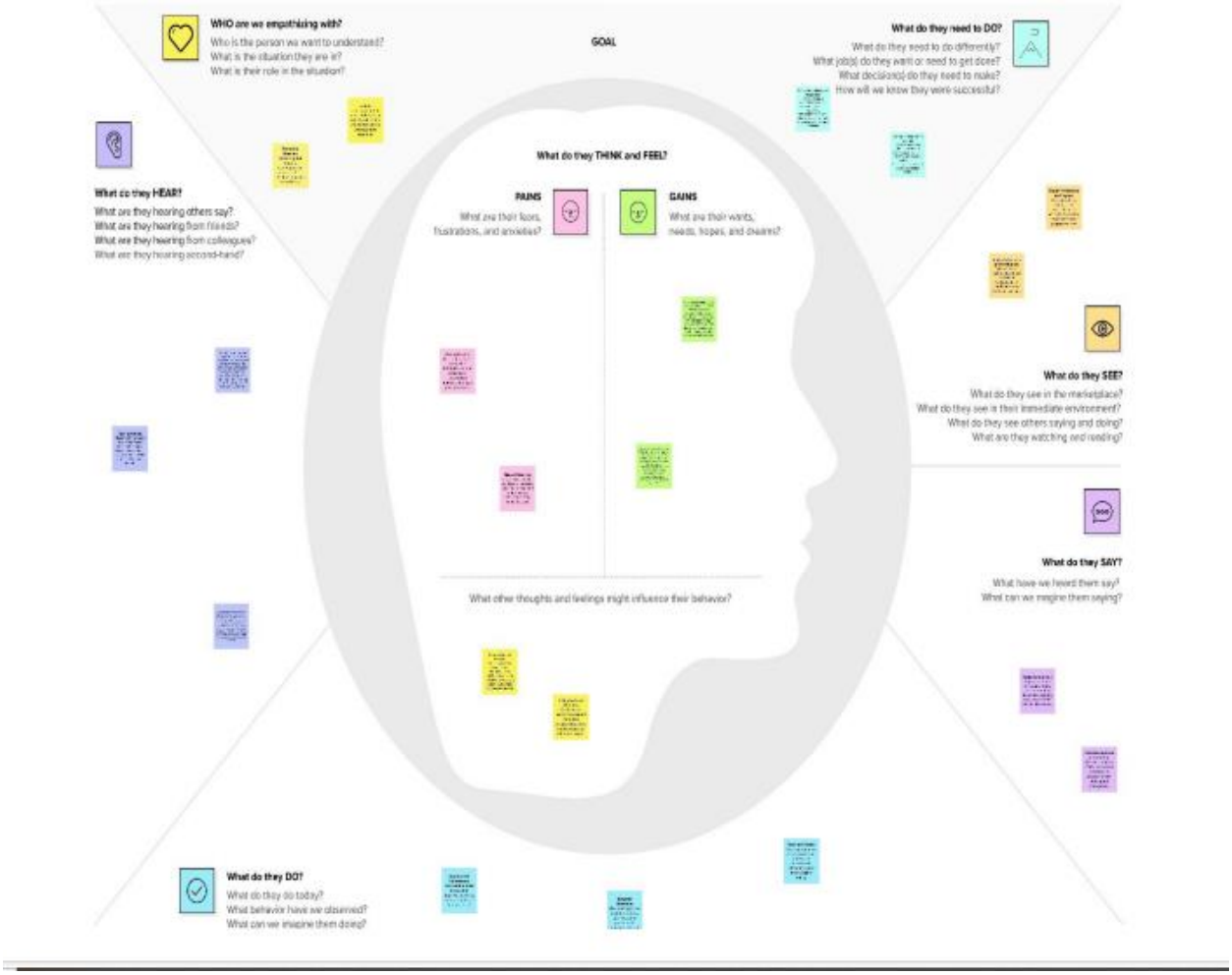
2. IDEATION AND PROPOSED SOLUTION

2.1. Problem Statement Solution

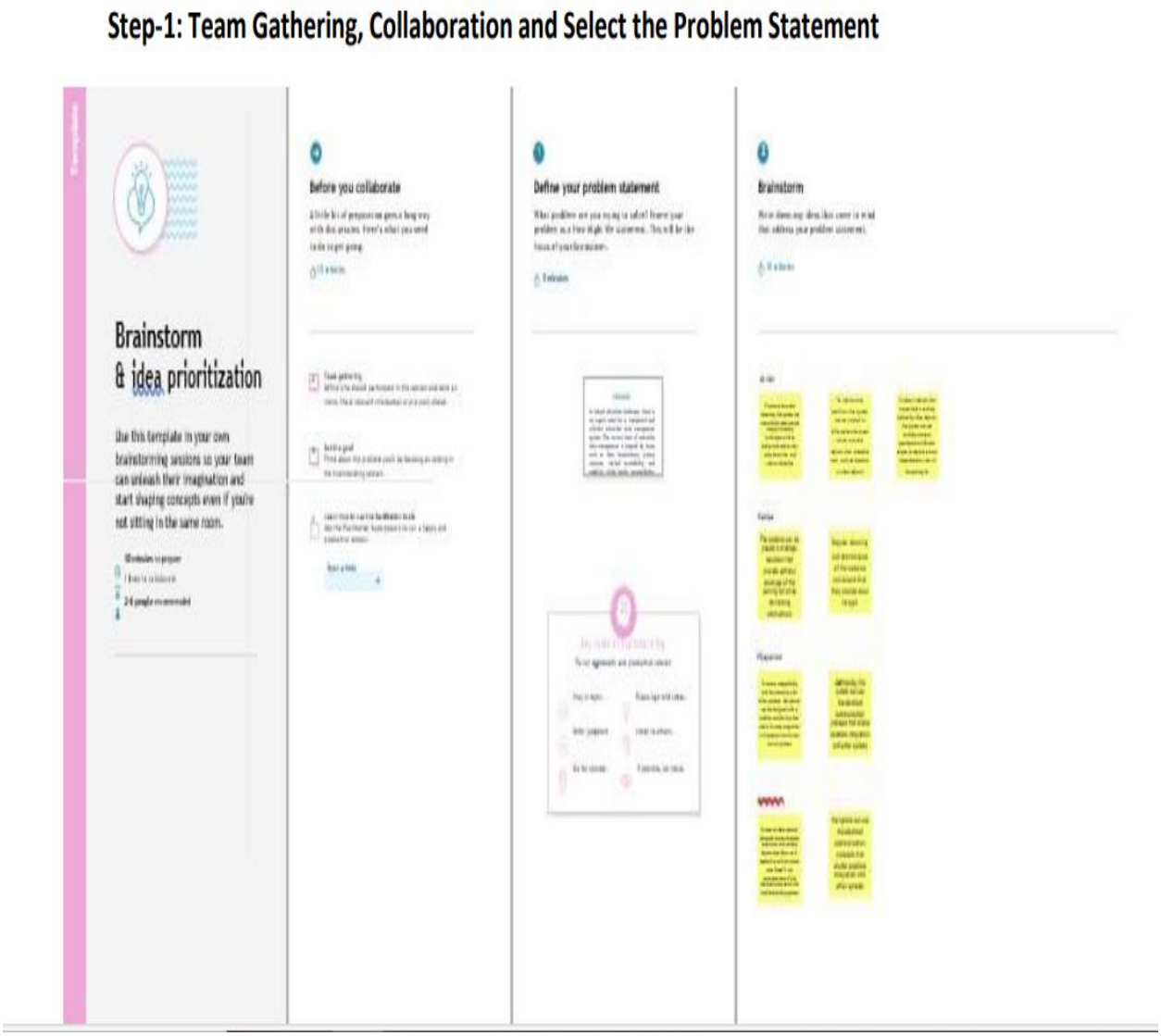
Data privacy is a critical issue in blockchain technology because of the transparency and immutability of the data stored on the blockchain. Once a transaction is added to the blockchain, it becomes permanent and cannot be altered or deleted. Supply chains, intellectual property, government operations, charity, voting, and crowdfunding are just a few of the pressing problems

that blockchain has the potential to address. It can also process transactions and eliminate inter

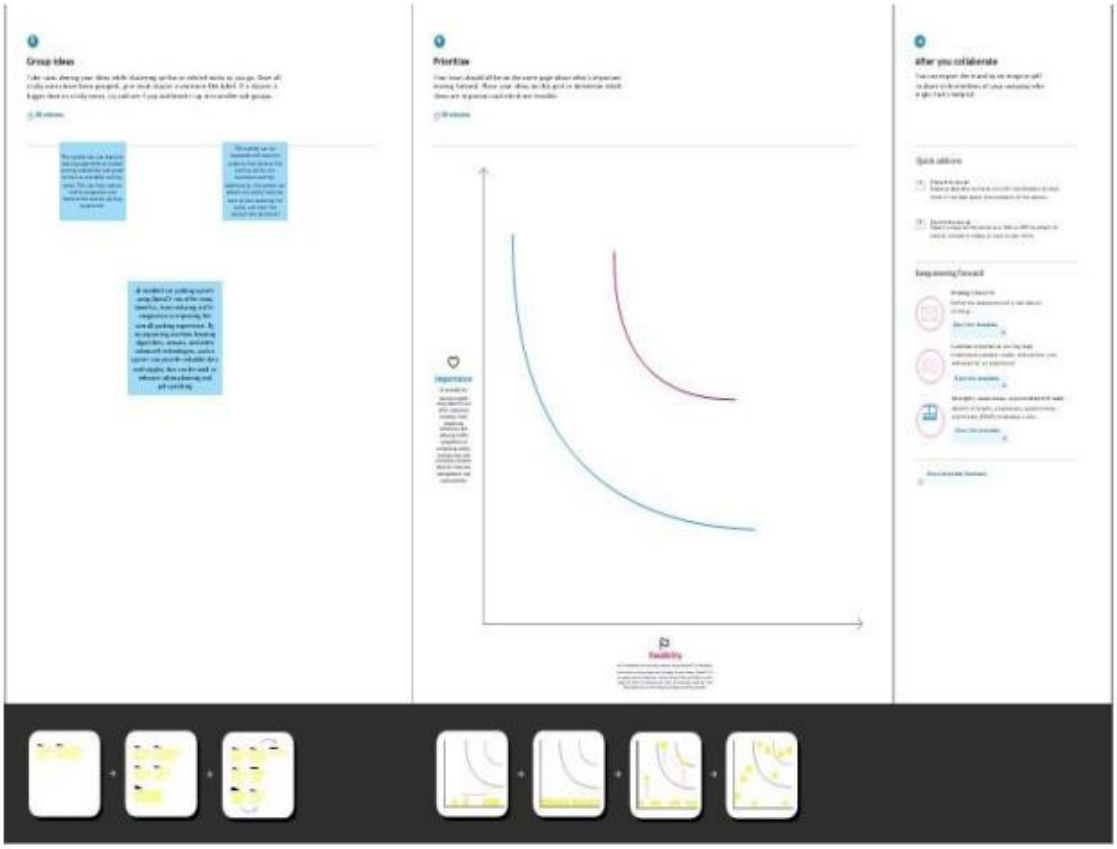
2.2. Empathy Map Canvas



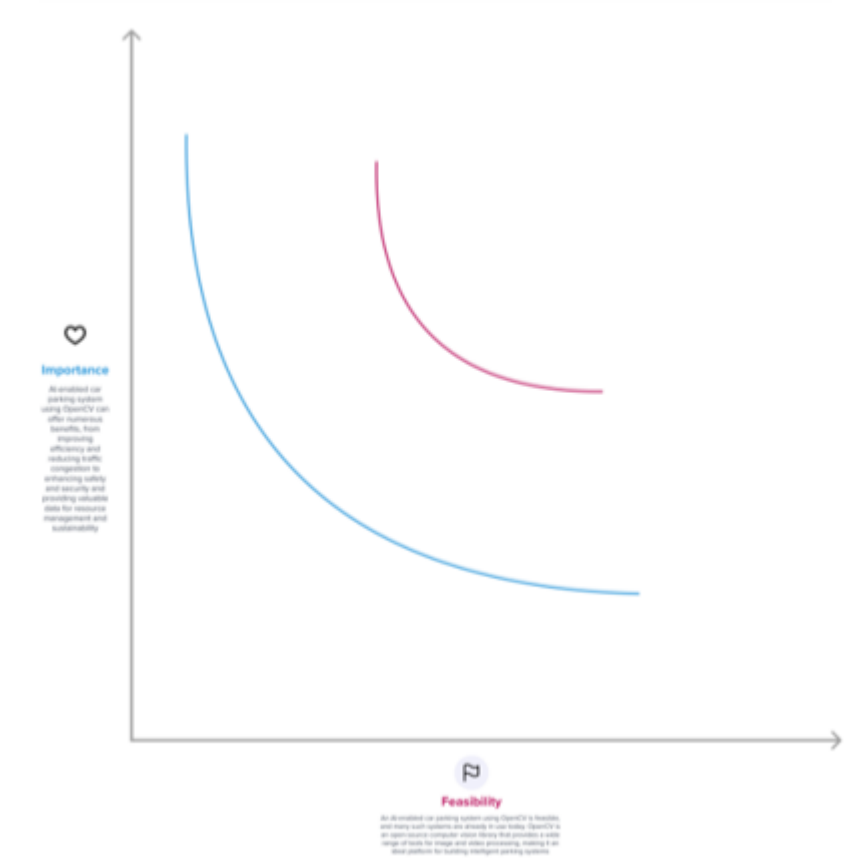
2.3. Ideation & Brainstorming



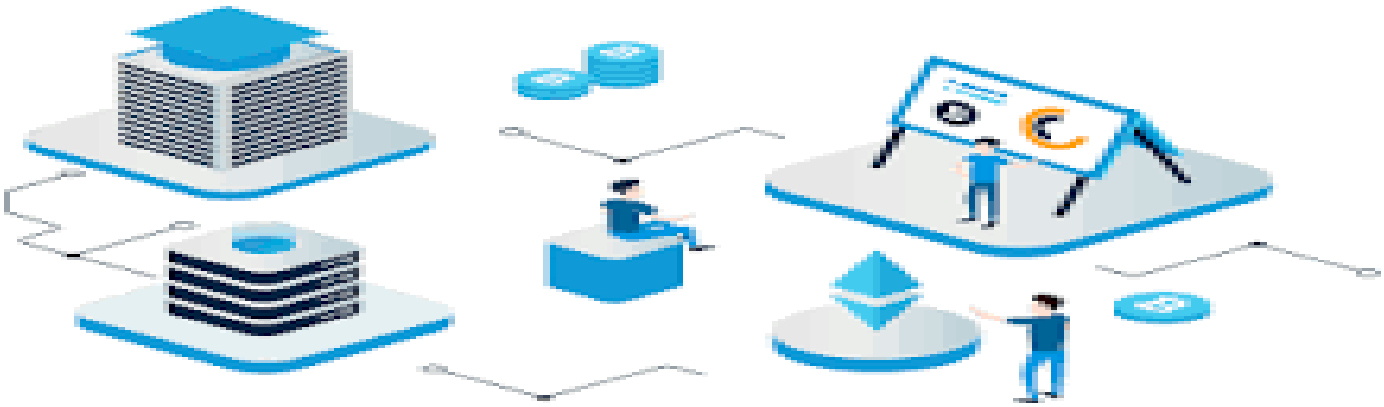
Step-2: Brainstorm, Idea Listing and Grouping



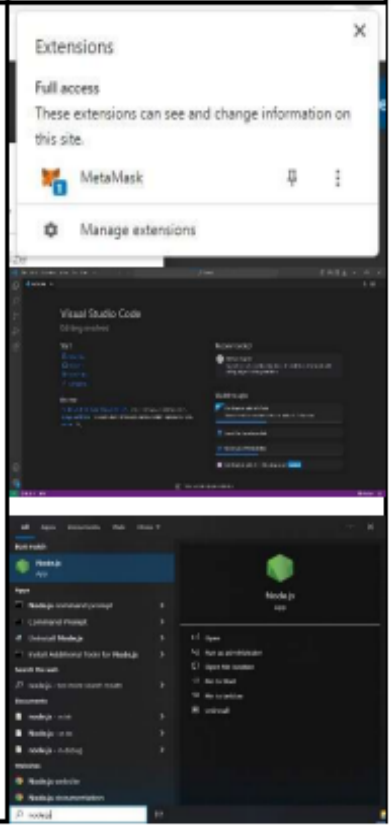
Step-3: Idea Prioritization



2.4. Proposed solution



Blockchain can be used in education to achieve greater transparency, credibility, accessibility, and traceability across various use cases like digital credentials, student record management, secure data sharing, content management, and more.

S.No.	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite:	

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 commands** 1.npm install 2.npm bootstrap 3. npm start	
5	{LOCALHOST IP ADDRESS	copy the address and open it to chrome so you can see the front end of your project.	

3.REQUIREMENT ANALYSIS

Functional Requirements	Non-Functional Requirements
They define a system or its component.	They define the quality attribute of a system
It specifies, “What the system should do?”	It specifies, “How should the system fulfill the functional requirements?”
User specifies functional requirement.	Non-functional requirement is specified by technical peoples. Architect, Technical leaders and software developers.
It is mandatory to meet these requirements.	It is not mandatory to meet these requirements.
It is captured in use case.	It is captured as a quality attribute.
Defined at a component level.	Applied to a whole system.
Helps you to verify the functionality of the software.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.

3.1.Functional Requirements

Following are the functional requirements of the proposed solution.

1. Authentication of a user when he/she tries to log into the system.

2. System shutdown in the case of a cyber attack.
3. Verification email is sent to user whenever he/she registers for the first time on some software system.

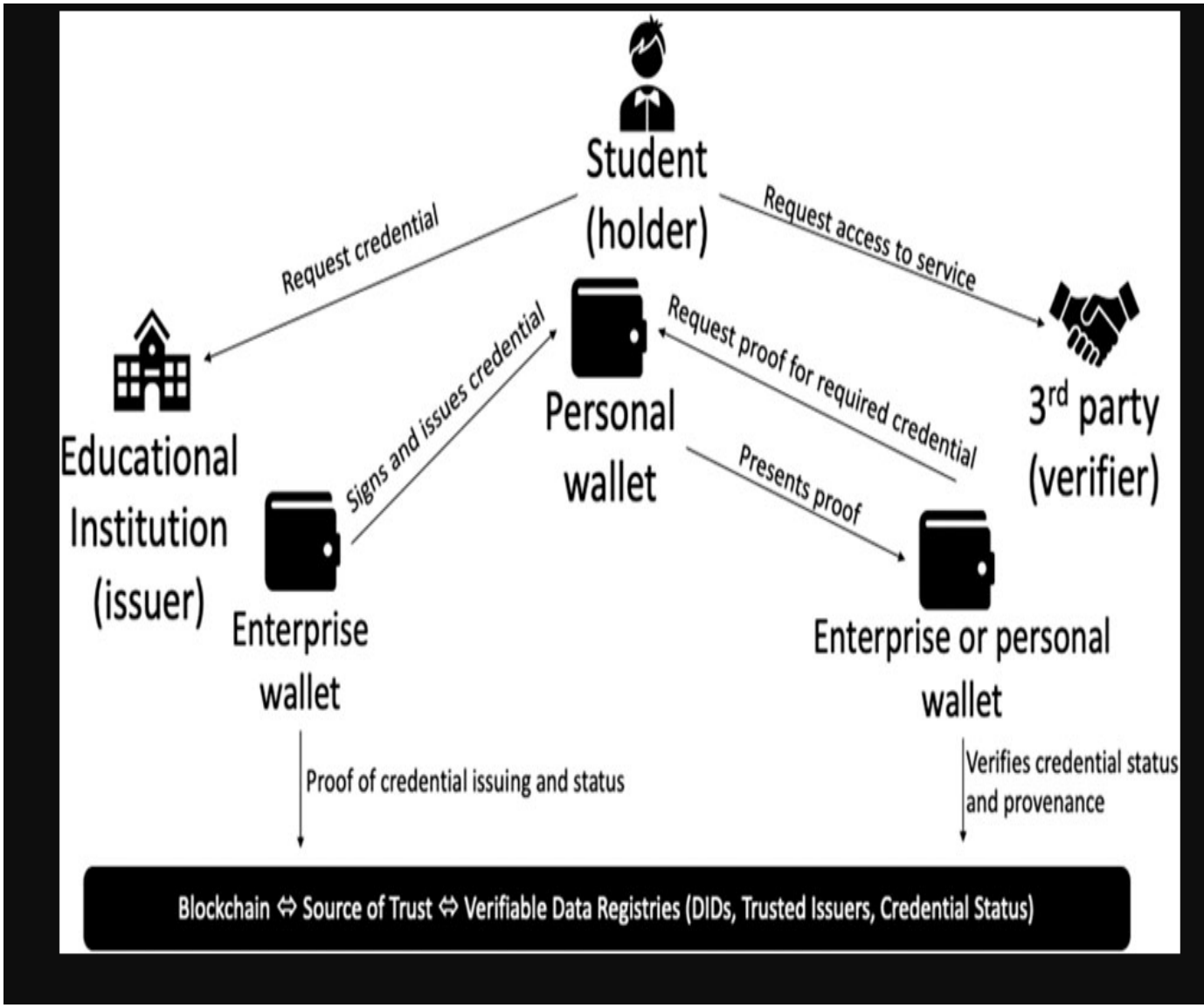
3.2.Non-Functional Requirements

Following are the non-functional requirements of the proposed solution.

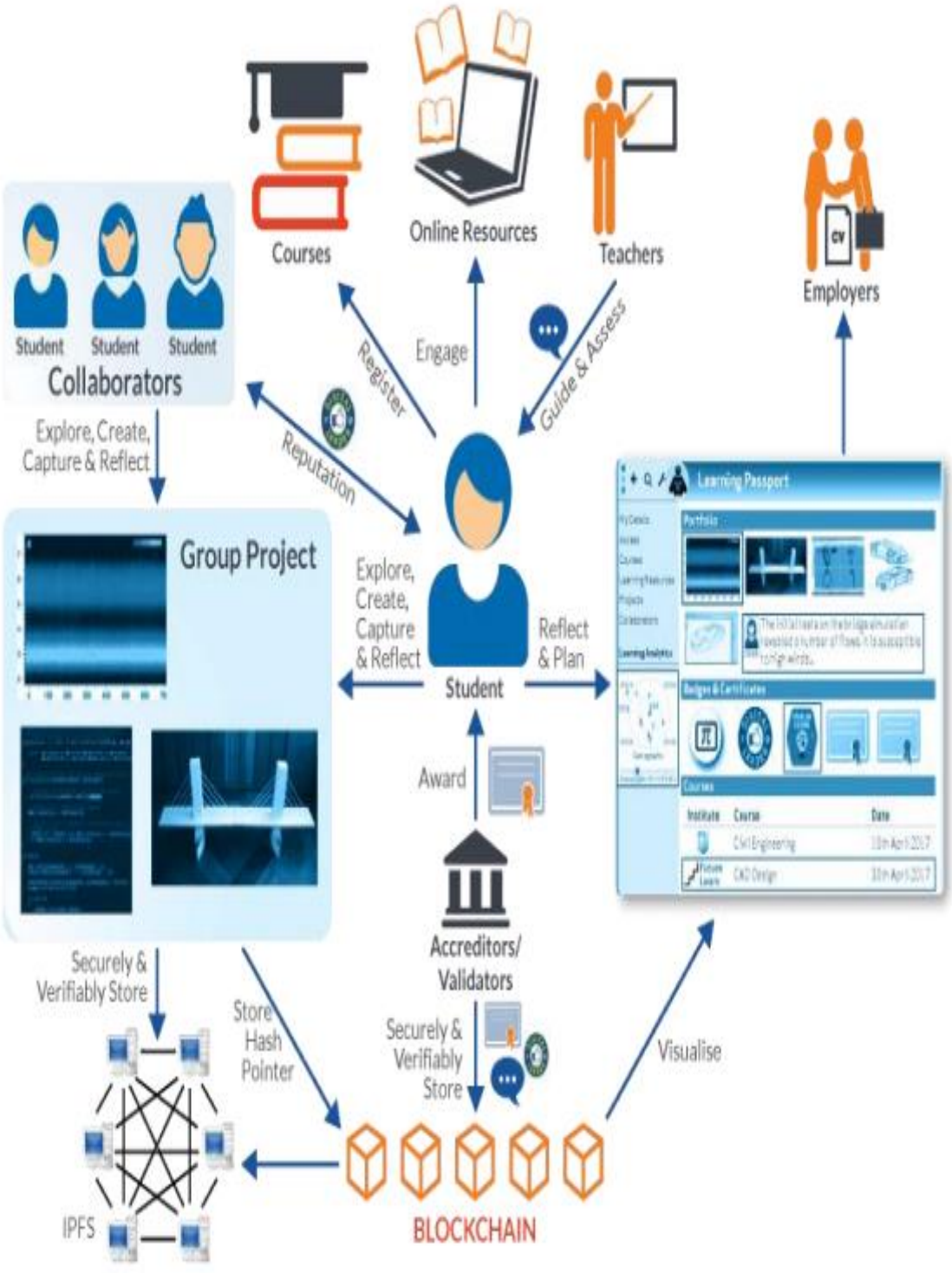
1. Emails should be sent with a latency of no greater than 12 hours.
2. Each request should be processed within 10 seconds.
3. The site should load in 3 seconds when the number of simultaneous users are > 10000

4.PROJECT DESIGN

4.1Data Flow Diagram



4.2 Solution & Technical Architecture



- Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

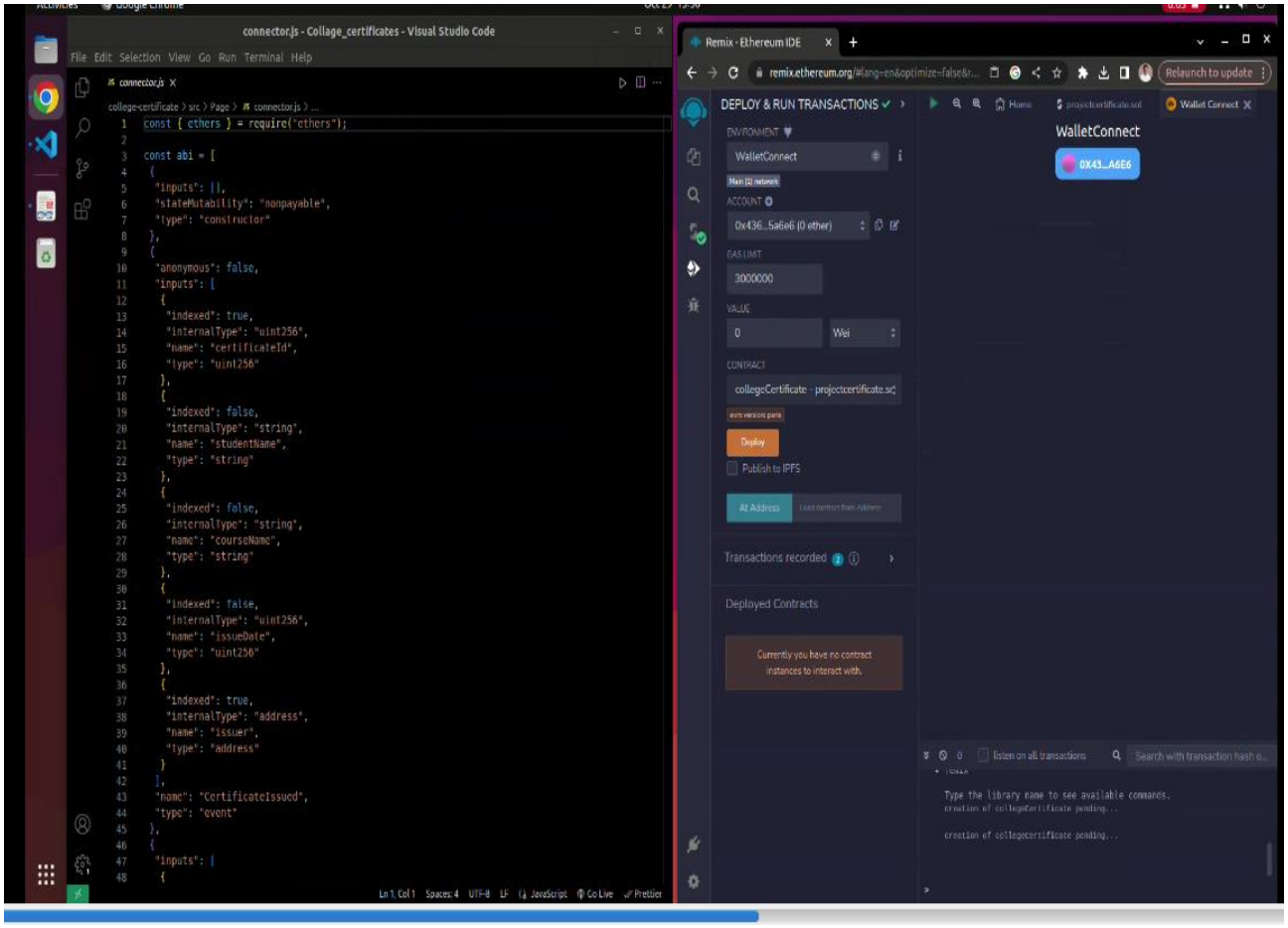
• Define features, development phases, and solution requirements.

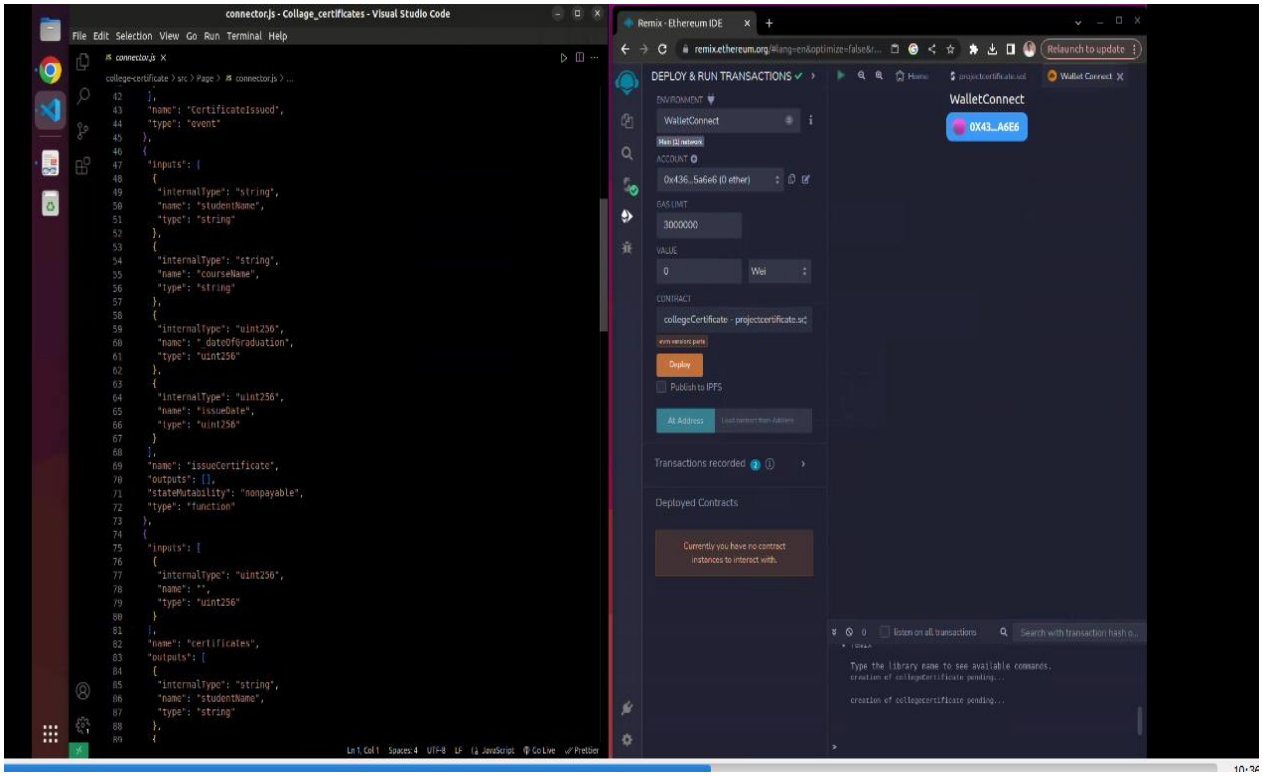
• Provide specifications according to which the solution is defined, managed, and delivered.

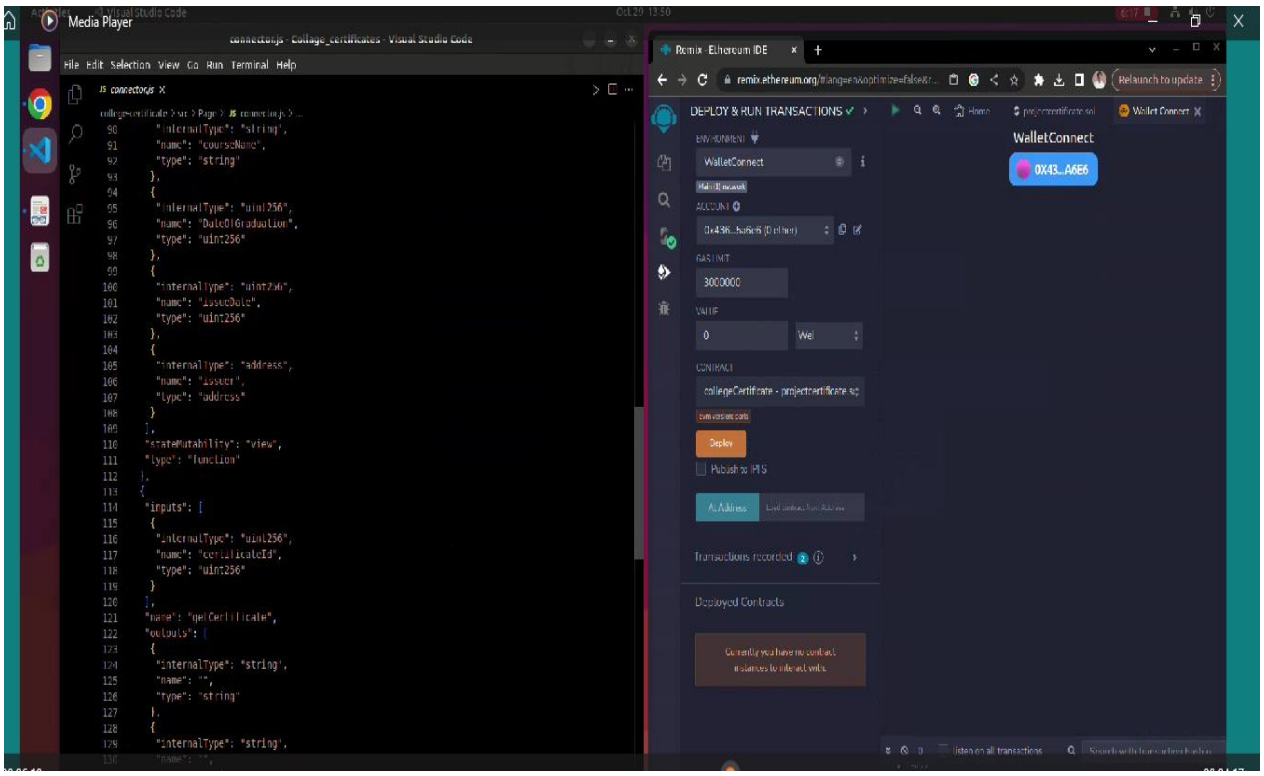
4.3. User Stories

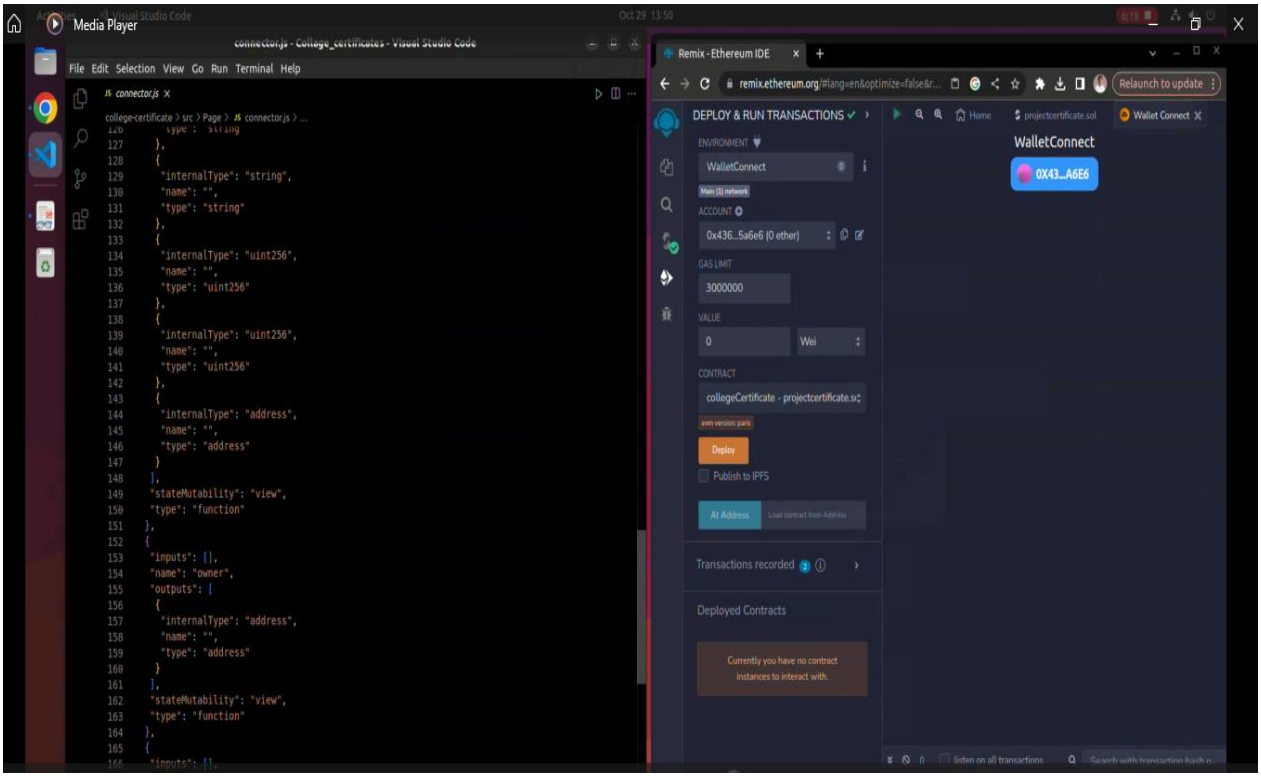
Use case	Application	Blockchain architecture	Consensus protocol	Maintainer of blockchain
V2X	On-road infotainment and location-dependent services	Permissionless	PoW/PoS	Third-party
		Permissioned	PBFT/Raft	RSU and edge nodes
Cloud/edge computing	Data auditing, data sharing, and data searching	Permissionless	PoW/PoS	Third-party
		Permissioned	PBFT/Raft	User and cloud/edge server

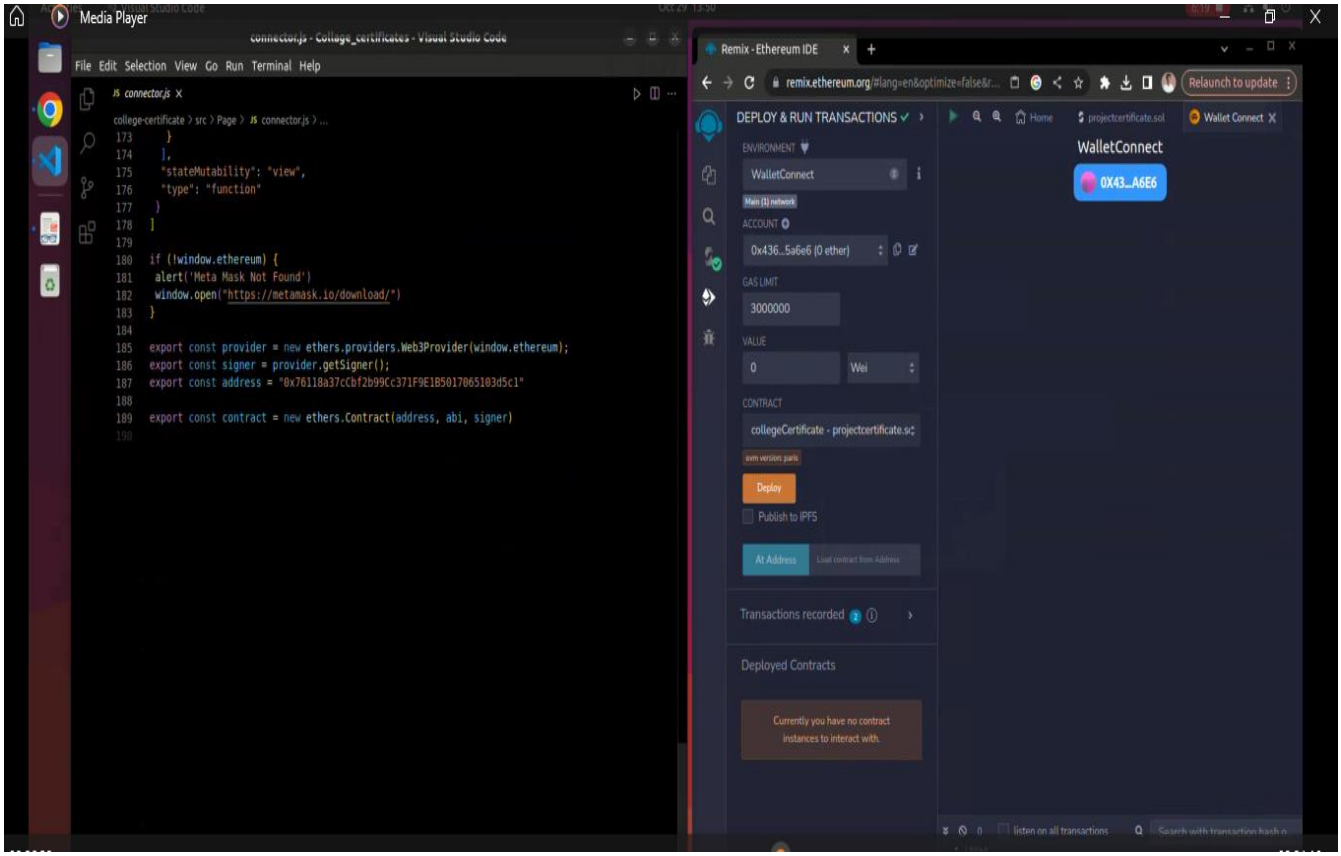
5.CODING & SOLUTIONING




connector.js - Collage_certificates - Visual Studio Code
File Edit Selection View Go Run Terminal Help
connector.js
college-certificate > src > Page > connector.js > ...
"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"
Ln 1, Col 1 Spaces: 4 UTF-8 LF JavaScript Go Live Prettier
Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&r...
Relaunch to update
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
WalletConnect
Main (1) network
ACCOUNT
0x436...5a6e6 (0 ether)
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
collegeCertificate - projectcertificate.sol
Deploy
Publish to IPFS
At Address
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.
Wallet Connect
WalletConnect
0X43...A6E6
Listen on all transactions
Search with transaction hash o...
Type the library name to see available commands.
creation of collegeCertificate pending...
creation of collegeCertificate pending...


Media Player
connector.js - College_certificates - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Remix - Ethereum IDE
DEPLOY & RUN TRANSACTIONS
WalletConnect
0X43...A6E6
Relaunch to update
Deploy
Publish to IPFS
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.


Media Player
connector.js - College_certificates - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Remix - Ethereum IDE
DEPLOY & RUN TRANSACTIONS
WalletConnect
0X43...A6E6
Relaunch to update
Deploy
Publish to IPFS
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.



7.ADVANTAGES & DISADVANTAGES

Advantages:

• **Immutability.** Blockchain supports immutability, meaning it is impossible to erase or replace recorded data. Therefore, the blockchain prevents data tampering within the network

Traditional data do not exhibit immutability. The conventional database uses CRUD (create, read, update and delete) at the primary level to ensure proper application operation, and the CRUD model enables easy erasing and replacing of data. Such data can be prone to manipulation by rogue administrators or third-party hacks.

• **Transparency.** Blockchain is decentralized, meaning any network member can verify data recorded into the blockchain. Therefore, the public can trust the network.

On the other hand, a traditional database is centralized and does not support transparency. Users cannot verify information whenever they want, and the administration makes a selected set of data public. Still, however, individuals cannot verify the data.

• **Censorship.** Blockchain technology is free from censorship since it does not have control of any single party. Therefore, no single authority (including governments) can interrupt the operation of the network.

Meanwhile, traditional databases have central authorities regulating the operation of the network, and the authority can exercise censorship. For instance, banks can suspend users' accounts.

• **Traceability.** Blockchain creates an irreversible audit trail, allowing easy tracing of changes on the network.

The traditional database is neither transparent nor immutable; hence, no permanent trail is guaranteed.

Disadvantages:

• **Speed and performance.** Blockchain is considerably slower than the traditional database because blockchain technology carries out more operations. First, it performs signature verification, which involves signing transactions cryptographically. Blockchain also relies on a consensus mechanism to validate transactions. Some consensus mechanisms, such as proof of

work, have a low transaction throughput. Finally, there is redundancy, where the network requires each node to play a crucial role in verifying and storing each transaction.

• **High implementation cost.** Blockchain is costlier compared to a traditional database. Additionally, businesses need proper planning and execution to integrate blockchain into their process.

• **Data modification.** Blockchain technology does not allow easy modification of data once recorded, and it requires rewriting the codes in all of the blocks, which is time-consuming and expensive. The downside of this feature is that it is hard to correct a mistake or make any necessary adjustments.

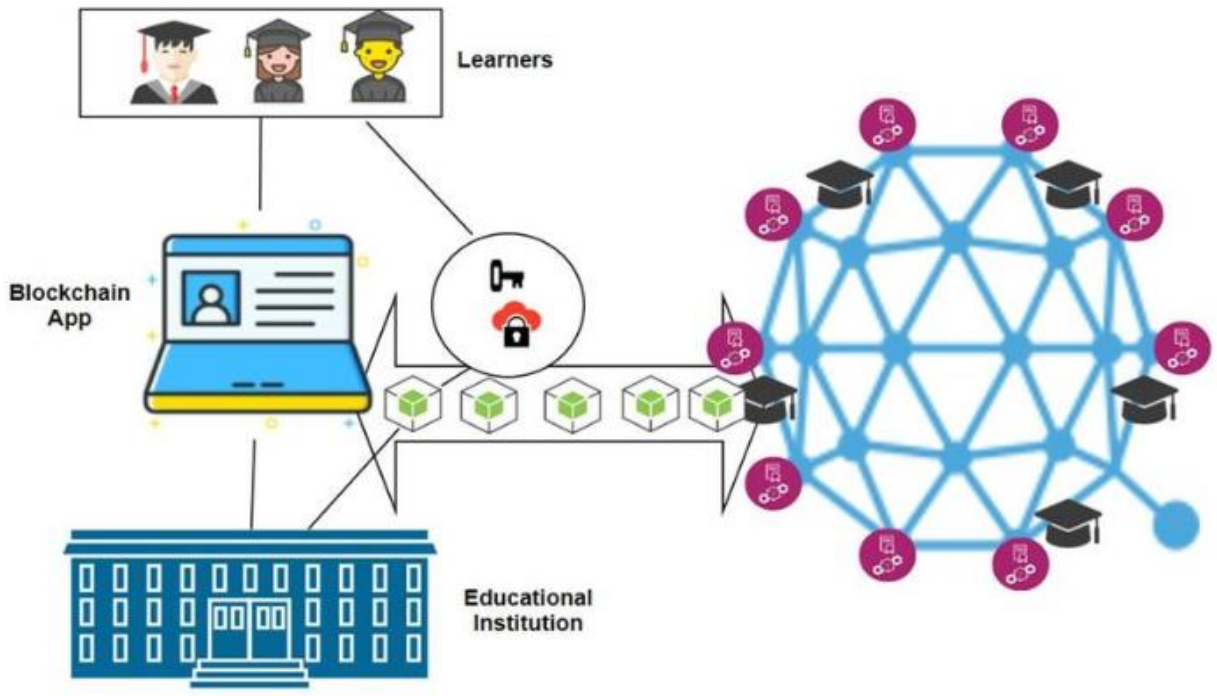
One solution doesn't fit all requirements, and this is the same with blockchain technology. There is a lot of buzz in the industry about blockchain and Web3, and many organizations are looking to move from Web 2.0 to Web3, but this is not a straightforward "lift-and-shift" type of solution. Organizations should do their due diligence and conduct a deep dive analysis to see if the blockchain technology fits their needs and then plan the development or migration to Web3 accordingly.

8.CONCLUSION

blockchain-based DM for 6G and highlighted its benefits of decentralization and transparency. By identifying efficiency and privacy challenges, we focused on DM architecture design, the AA of DM stakeholders, and blockchain-based data processing.

To explore potential solutions that balance transparency, efficiency, and privacy in decentralized blockchain-based DM, further research can be directed to the following open issues. First, the impact of network virtualization on DM architecture design should be discussed. Blockchain-based DM requires a flexible and versatile architecture with efficient consensus protocols, inter-chain operability, and fast service-oriented configurations. Second, lightweight and distributed AA with dynamic updates should be designed in order to strike a balance between AA privacy and accountability for blockchain-based DM. Third, an executable privacy model that can accommodate a wide range of privacy requirements in different DM operations should be achieved. Modular integration of privacy-preserving data-processing techniques should be explored under the privacy models.

9.FUTURESCOPE



10. APPENDIX

GITHUB PROJECT LINK: