DISTRIBUTED LEDGER FRAMEWORK

by

Sudigali Ala Sai   421260

Sannekanti Sathvik   421251

Under the guidance of

Dr. Nagesh Bhattu Sristy

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534101, INDIA

MAY 2023

DISTRIBUTED LEGDER FRAMEWORK

Thesis submitted to
National Institute of Technology Andhra Pradesh
for the award of the degree

of

Bachelor of Technology

by

Sudigali Ala Sai 421260
Sannekanti Sathvik 421251

Under the guidance of

Dr. Nagesh Bhattu Sristy

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534101, INDIA

MAY 2023

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sudigali Ala Sai	Sannekanti Sathvik
421260	421251
Date:____________	Date:____________

CERTIFICATE

It is certified that the work contained in the thesis titled "**Distributed Ledger Framework**" by "Sudigali Ala Sai, bearing Roll No: 421260" and "Sannekanti Sathvik,bearing Roll No: 421251" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature
Dr. Nagesh Bhattu Sristy
DCSE
N.I.T. Andhra Pradesh
May, 2023

Acknowledgement

I would like to express my sincere gratitude to my supervisor for their unwavering support and encouragement throughout my research journey. Their guidance and motivation have been invaluable in helping me achieve my goals.

LIST OF FIGURES

Abstract

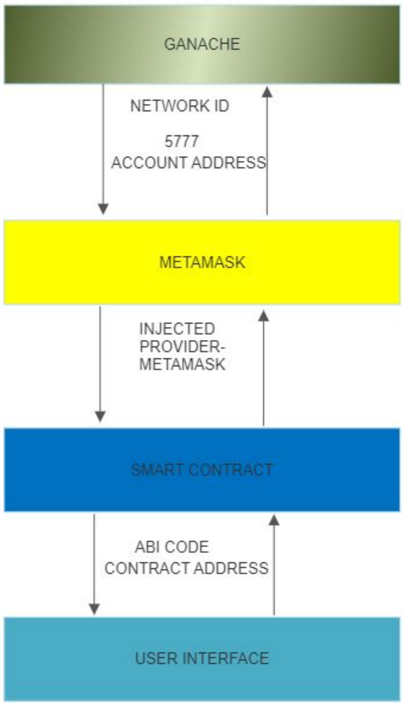
This project aims to develop a distributed ledger framework using Ethereum blockchain to securely store and manage academic marks. The proposed system offers a decentralized and transparent solution to traditional mark management systems, which often suffer from security vulnerabilities and data tampering.The framework will utilize smart contracts on the Ethereum blockchain to automate the mark storage and verification process.The proposed system will be designed with a focus on security, privacy, and scalability. The use of blockchain technology ensures that marks cannot be altered or deleted.The implementation of the proposed framework has the potential to revolutionize the way academic marks are stored and managed. By using a distributed ledger on the Ethereum blockchain, the system will provide a transparent and secure solution to traditional mark management systems.This project involves developing a distributed ledger framework using Ethereum blockchain and integrating it with the Metamask wallet. The proposed system will provide a secure and decentralized solution for managing academic marks, while also allowing users to easily interact with the blockchain using their Metamask wallet.The system will utilize smart contracts on the Ethereum blockchain to automate the mark storage and verification process, while also ensuring the privacy and security of sensitive data. The integration of Metamask will enable users to securely manage their private keys and sign transactions, improving the overall usability and user experience of the system.The proposed framework with Metamask integration has the potential to revolutionize the way academic marks are managed and verified. By combining the security and transparency of the blockchain with the ease of use of Metamask, the system will provide a powerful and accessible tool for managing academic marks in a decentralized and secure manner.

TABLE OF CONTENTS

Contents

1 Introduction

The use of blockchain and other distributed ledger technologies has been rapidly expanding in recent years, with applications in finance, supply chain management, and more. One area where distributed ledger frameworks could have a significant impact is in university management systems. Distributed ledger frameworks are based on a consensus algorithm, which ensures that all participants agree on the state of the ledger. This eliminates the need for a trusted third party to manage the ledger, and allows multiple parties to access, view, and verify the same data.For example, student records could be stored on a decentralized network, making it harder for hackers to compromise the data.



While there are still challenges to be overcome in implementing a distributed ledger framework in a university management system, such as scalability and interoperability between different systems, the potential benefits are significant. A distributed ledger framework could improve the security, transparency, and efficiency of university management systems, making it easier for universities to manage their records and better serve their students.

2 Literature Review

Blockchain technology has gained significant attention in recent years due to its ability to provide secure and transparent solutions for various applications. In the field of education, blockchain has been proposed as a potential solution for managing academic records and credentials.

2.1 Blockchain

Blockchain technology has gained significant attention in recent years due to its ability to provide secure and transparent solutions for various applications.

2.1.1 Problem statement

The management and verification of academic marks is a critical issue faced by educational institutions worldwide. Current systems for managing academic marks are centralized and prone to errors, fraud, and data tampering. In addition, existing systems lack transparency and accessibility, which can lead to delays in the verification of marks and certificates.To address these issues, there is a need for a decentralized, secure, and transparent solution for managing academic marks. The proposed solution should ensure the privacy and security of sensitive data, while also providing easy access to verified marks. The use of blockchain technology, specifically Ethereum blockchain, has the potential to provide such a solution.

2.1.2 smart contract

A smart contract is a self-executing contract with the terms of the agreement between the buyer and seller being directly written into lines of code. It is a computer program that automatically executes the terms of a contract when certain conditions are met. Smart contracts are stored on a blockchain network, such as Ethereum, which provides a secure and transparent platform for executing the contract.

2.1.3 Types of Blockchain

There are generally three types of blockchain: public, private, and consortium.A public blockchain is a decentralized and open network that allows anyone to participate, view and validate transactions.A consortium blockchain is a hybrid between public and private blockchains. It is a permissioned network that is controlled by a group of organizations that agree to work together.

2.2 Benefits of Using Ledger Framework

Blockchain technology has numerous potential applications across various industries and sectors.

Blockchain technology has many potential uses in various industries and sectors. One of the most well-known applications of blockchain is cryptocurrency, such as Bitcoin and Ethereum, which uses blockchain technology to enable secure and transparent transactions without the need for intermediaries. Another notable application of blockchain is in supply chain management, where it can be used to track and authenticate the supply chain of products, ensuring transparency and reducing the risk of fraud. Blockchain can also be used for digital identity management, creating secure and tamper-proof digital identities that can be used for authentication and verification purposes. Smart contracts are another potential use case for blockchain technology, allowing for the automation of complex transactions and reducing the need for intermediaries.

2.3 Drawbacks of Ledger Framework

Although distributed ledger frameworks such as blockchain offer many potential benefits, there are also some notable drawbacks to consider. One of the biggest challenges facing distributed ledger frameworks is scalability. As the size of the network grows, so does the amount of data that needs to be processed and stored, which can slow down the network and increase transaction fees. While blockchain technology is often touted as being highly secure, there are still some security risks.

2.4 Future Directions

The future of ledger frameworks such as blockchain is likely to be characterized by continued development and innovation, as well as increasing integration with other technologies and applications.

Scalability is likely to remain a key challenge for distributed ledger frameworks, and we can expect to see continued efforts to address this issue. This could include the development of new consensus algorithms, such as proof-of-stake, as well as the integration of off-chain scaling solutions such as the Lightning Network.As blockchain and other ledger frameworks become more mainstream, we are likely to see increasing integration with other technologies and applications. This could include the integration of blockchain with artificial intelligence, Internet of Things (IoT) devices, and other emerging technologies, enabling new use cases and applications.

Finally, the future of ledger frameworks is likely to be shaped by ongoing regulatory and compliance issues. As the technology becomes more widely adopted, it will be important for regulatory frameworks to catch up, ensuring that the technology can be used in a safe and secure manner while still fostering innovation and growth.

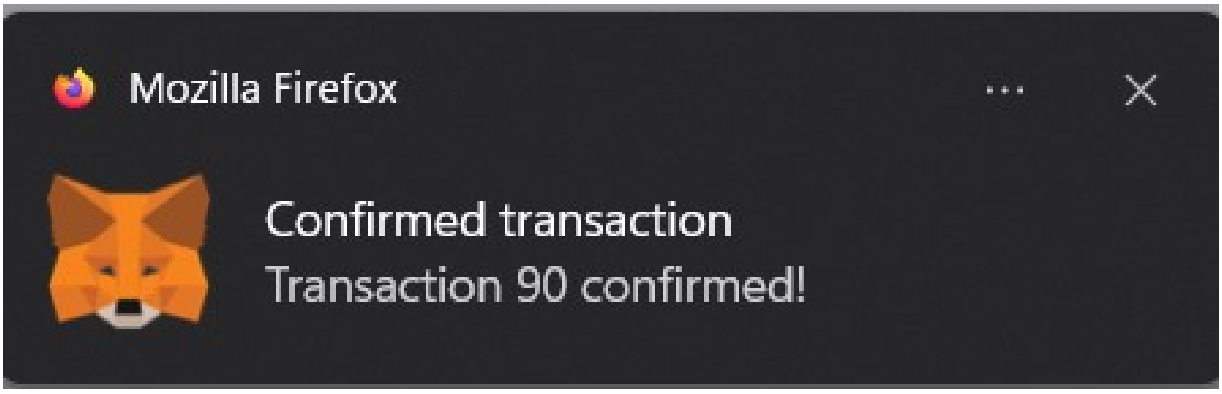
3 Metmask and Ganache

MetaMask is a popular cryptocurrency wallet and browser extension that allows users to interact with the Ethereum blockchain. It provides a simple and user-friendly interface for managing Ethereum wallets, sending and receiving transactions, and interacting with decentralized applications (dApps) that run on the Ethereum network.

MetaMask acts as a bridge between the user's web browser and the Ethereum blockchain, allowing users to interact with dApps without the need to run a full Ethereum node. This makes it easier for users to access and use dApps, as they do not need to have technical knowledge or set up complex infrastructure.

One of the key features of MetaMask is its ability to store and manage multiple Ethereum accounts, allowing users to easily switch between accounts and manage their funds. It also provides a secure and encrypted environment for storing private keys, ensuring that users' funds are kept safe.

MetaMask is widely used by developers and users of Ethereum-based dApps, as it provides a convenient way to interact with the blockchain and manage accounts. It is also supported by a wide range of dApps and services, making it easy for users to find and use new applications.



Overall, MetaMask is a powerful tool for anyone looking to interact with the Ethereum blockchain and use decentralized applications. Its user-friendly interface and wide range of features have made it a popular choice for users and developers alike. Ganache provides a local blockchain that runs on the developer's machine, allowing them to simulate a full Ethereum network without the need for a public test network or the use of real Ether. This makes it easier for developers to test and debug their applications in a controlled environment before deploying them to a live network.

Ganache comes with a user interface that provides detailed information about the blockchain, including account balances, transaction history, and contract deployment status. It also allows developers to create and manage multiple accounts, each with their own balance of test Ether.Overall, Ganache is a powerful tool for Ethereum developers, providing a sandboxed environment for testing and developing blockchain applications. Its user-friendly interface, advanced features, and support for multiple network scenarios make it a popular choice among developers looking to build and deploy decentralized applications on the Ethereum blockchain.

4 Solidity Code

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

contract GradeSubmission {
    struct Grade {
        uint256 marks;
        string grade;
        bool approved;
    }

    mapping(uint256 => Grade) grades;
    address public faculty;
    address public higherAuthority;

    modifier onlyFaculty() {
        require(msg.sender == faculty,
         "Only the faculty can call this function.");
        _;
    }

    modifier onlyHigherAuthority() {
        require(msg.sender == higherAuthority,
         "Only the higher authority can call this function.");
        _;
    }

    constructor(address _faculty, address _higherAuthority) {
        faculty = _faculty;
        higherAuthority = _higherAuthority;
    }

    function submitGrade(uint256 rollNumber, uint256 marks) public onlyFaculty {
        require(marks <= 100, "Marks should be less than or equal to 100");
        require(bytes(grades[rollNumber].grade).length == 0,
         "Grade already submitted for this roll number");
        string memory grade = getGrade(marks);
        grades[rollNumber] = Grade(marks, grade, false);
    }
```

```
function getGrade(uint256 marks) internal pure returns (string memory) {
    if (marks >= 90) {
        return "A+";
    } else if (marks >= 80) {
        return "A";
    } else if (marks >= 70) {
        return "B+";
    } else if (marks >= 60) {
        return "B";
    } else if (marks >= 50) {
        return "C+";
    } else if (marks >= 40) {
        return "C";
    } else {
        return "F";
    }
}

function approveGrade(uint256 rollNumber) public onlyHigherAuthority {
    require(bytes(grades[rollNumber].grade).length > 0,
    "No grade submitted for this roll number");
    require(!grades[rollNumber].approved,
     "Grade already approved");
    grades[rollNumber].approved = true;
}

function viewMyGrade(uint256 rollNumber) public view returns
 (uint256, string memory, bool) {
    require(msg.sender == faculty || msg.sender == higherAuthority,
    "Only faculty or higher authority can view the grade.");
    require(bytes(grades[rollNumber].grade).length > 0,
    "No grade submitted for this roll number");
    return (grades[rollNumber].marks, grades[rollNumber].grade,
     grades[rollNumber].approved);
}
}
```

Smart contract code return in solidity language includes a mapping that maps roll numbers to their respective marks.The contract has two addresses: faculty and higherAuthority, which are set during the contract's deployment.The mapping named grades maps each roll number to its corresponding GradeSubmission struct. By using a mapping, the contract allows for efficient and easy storage and retrieval of student grades.The constructor function takes two arguments: the faculty and higherAuthority addresses. These addresses are set during deployment and cannot be changed later.The onlyFaculty and onlyHigherAuthority modifiers use the require statement to check if the caller of a function is the faculty or higherAuthority respectively. If the condition is not met, the function will revert and the transaction will not be executed.

The faculty can submit marks for students using the submitGrade function. This function takes two arguments: the roll number of the student and their marks. It also checks if the marks are less than or equal to 100 and whether a grade has already been submitted for that roll number.

The higherAuthority can approve grades using the approveGrade function. This function takes the roll number of the student as an argument and checks if a grade has been submitted for that roll number and whether the grade has already been approved. If both conditions are met, the function sets the approved flag for that grade to true.

The viewMyGrade function allows the faculty and higherAuthority to view the grades of a student. This function takes the roll number of the student as an argument and returns the marks and approval status for that student.The contract also includes two modifiers: onlyFaculty and onlyHigherAuthority, which restrict the access to certain functions to the faculty and higherAuthority respectively.

Overall, this smart contract is designed to provide a secure and transparent way for schools and universities to manage student grades. The use of modifiers and internal functions ensures that the contract can only be accessed by authorized parties, while the use of a mapping allows for efficient storage and retrieval of student grades.

5 Benefits over centralized database

5.1 Decentralization

A major advantage of using Solidity and smart contracts is the decentralization of data. Instead of storing data in a centralized database, data is stored on a blockchain network, which means that there is no single point of failure or control. This can increase the security, reliability, and transparency of the system.

5.2 Trustless Transactions

Transactions on a blockchain network using smart contracts are trustless, meaning they do not require trust between parties involved. Transactions are verified and validated by the network itself and are immutable, meaning they cannot be tampered with.

5.3 Transparent and Auditable

Transactions on a blockchain network are transparent and can be audited by anyone on the network. This can help increase accountability and transparency in the system.

5.4 Accessibitity

MetaMask is a browser extension that allows users to interact with blockchain networks and smart contracts, while Ganache is a personal blockchain that allows developers to test their smart contracts in a local environment. These tools can help make the development and deployment of smart contracts easier and more accessible.

5.5 Reduced cost

Smart contracts can be programmed to execute automatically based on pre-defined conditions. This can automate processes and reduce the need for human intervention, which can lead to greater efficiency and reduced costs.

Overall, using Solidity, smart contracts, MetaMask, and Ganache can provide a more secure, efficient, and transparent alternative to traditional centralized databases. By leveraging the benefits of blockchain technology, organizations can increase trust, reduce costs, and improve efficiency in their operations.

6 Applications of Blockchain

With blockchain, marks and grades can be stored securely and transparently on a decentralized network, providing a tamper-proof and auditable record of academic achievements.

6.1 Digital Credentials

Blockchain can be used to issue and verify digital credentials, such as diplomas, certificates, and transcripts. These credentials can be stored securely on a blockchain network and accessed by students, employers, and other relevant parties. This can reduce fraud, increase trust, and simplify the verification process.

6.2 Data Sharing

Blockchain can be used to facilitate data sharing between educational institutions, students, and other relevant parties. This can help improve collaboration and reduce duplication of effort, while also ensuring that data is shared securely and transparently.

6.3 land ownership and transactions

Blockchain technology provides a secure and transparent way to record and manage land ownership. Every transaction is recorded on the blockchain, making it nearly impossible to tamper with or manipulate. This can increase transparency and reduce the risk of fraud and corruption.

7 More about Blockchain

A blockchain is a public database that is updated and shared across many computers in a network.”Block” refers to data and state being stored in consecutive groups known as ”blocks”. If you send ETH to someone else, the transaction data needs to be added to a block to be successful.”Chain” refers to the fact that each block cryptographically references its parent. In other words, blocks get chained together. The data in a block cannot change without changing all subsequent blocks, which would require the consensus of the entire network.Every computer in the network must agree upon each new block and the chain as a whole. These computers are known as ”nodes”. Nodes ensure everyone interacting with the blockchain has the same data. To accomplish this distributed agreement, blockchains need a consensus mechanism.

In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on. Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer. Additionally, any participant can broadcast a request for this computer to perform arbitrary computation. Whenever such a request is broadcast, other participants on the network verify, validate, and carry out (”execute”) the computation. This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.

Ether (ETH) is the native cryptocurrency of Ethereum. The purpose of ETH is to allow for a market for computation. Such a market provides an economic incentive for participants to verify and execute transaction requests and provide computational resources to the network. Nodes communicate with each other to propagate information about the EVM state and new state changes. Any user can also request the execution of code by broadcasting a code execution request from a node. The Ethereum network itself is the aggregate of all Ethereum nodes and their communications. Ether (ETH) is the cryptocurrency used for many things on the Ethereum network. Fundamentally, it is the only acceptable form of payment for transaction fees, and after The Merge, ether is required to validate and propose blocks on Mainnet. Ether is also used as a primary form of collateral in the DeFi lending markets, as a unit of account in NFT marketplaces, as payment earned for performing services or selling real-world goods, and more.The ether cryptocurrency supports a pricing mechanism for Ethereum’s computing power. When users want to make a transaction, they must pay ether to have their transaction recognized on the blockchain. These usage costs are known as gas fees, and the gas fee depends on the amount of computing power required to execute the transaction and the network-wide demand for computing power at the time.

Minting is the process in which new ether gets created on the Ethereum ledger. The underlying Ethereum protocol creates the new ether, and it is not possible for a user to create ether.Ether is minted as a reward for each block proposed and at every epoch checkpoint for other validator activity related to reaching consensus.

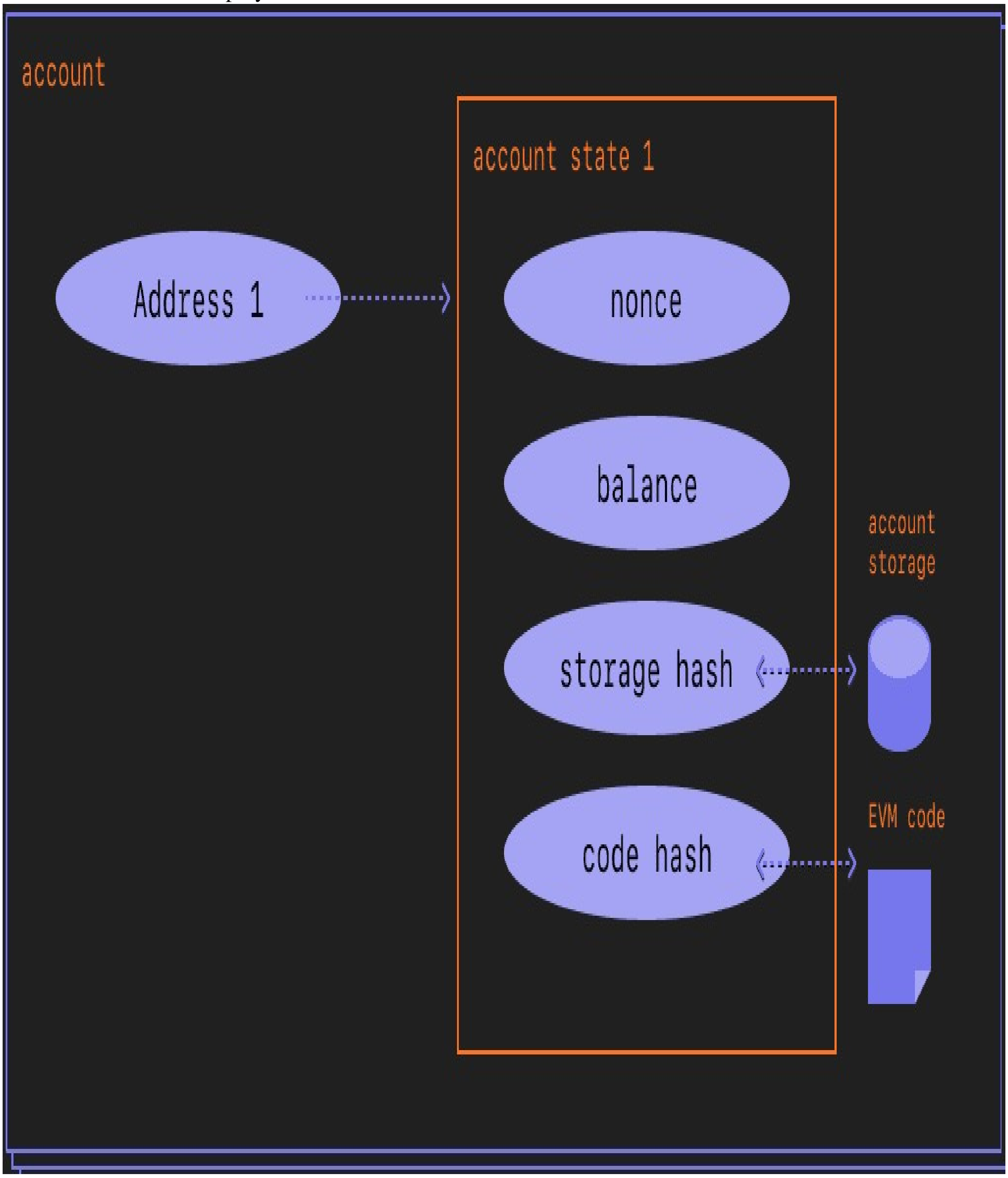
A decentralized application (dapp) is an application built on a decentralized network that combines a smart contract and a frontend user interface. On Ethereum, smart contracts are accessible and transparent – like open APIs – so your dapp can even include a smart contract that someone else has written.A smart contract is code that lives on the Ethereum blockchain and runs exactly as programmed. Once smart contracts are deployed on the network you can’t change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company.

Ethereum uses a proof-of-stake-based consensus mechanism that derives its crypto-economic security from a set of rewards and penalties applied to capital locked by stakers. This incentive structure encourages individual stakers to operate honest validators, punishes those who don’t, and creates an extremely high cost to attack the network.

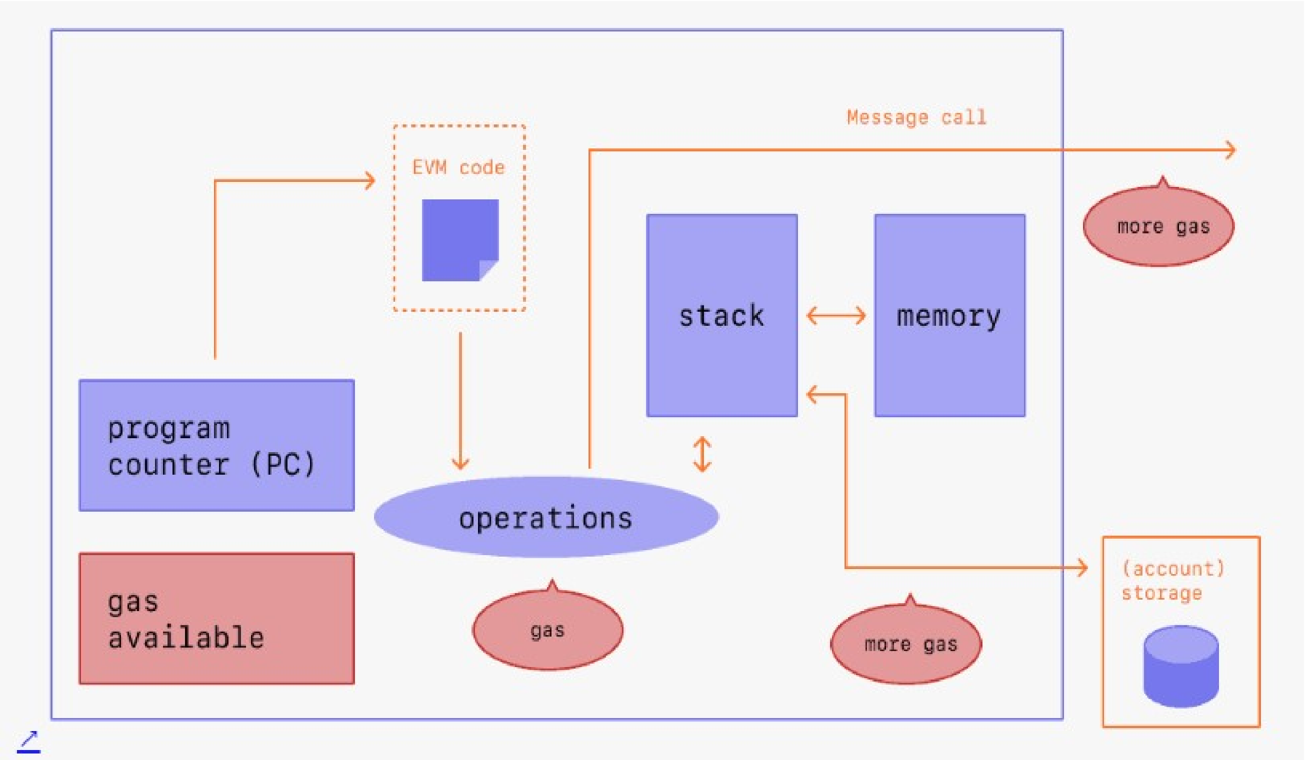
Your Ethereum account will work across the different networks, but your account balance and transaction history won't carry over from the main Ethereum network. For testing purposes, it's useful to know which networks are available and how to get testnet ETH to play around with. In general, for security considerations, it's not recommended to reuse mainnet accounts on testnets or vice versa.

8 Accounts and Transactions

An Ethereum account is an entity with an ether (ETH) balance that can send transactions on Ethereum. Accounts can be user-controlled or deployed as smart contracts.



Blocks are batches of transactions with a hash of the previous block in the chain. This links blocks together (in a chain) because hashes are cryptographically derived from the block data. This prevents fraud, because one change in any block in history would invalidate all the following blocks as all subsequent hashes would change and everyone running the blockchain would notice.Once a block is put together by a randomly selected validator on the network, it is propagated to the rest of the network; all nodes add this block to the end of their blockchain.



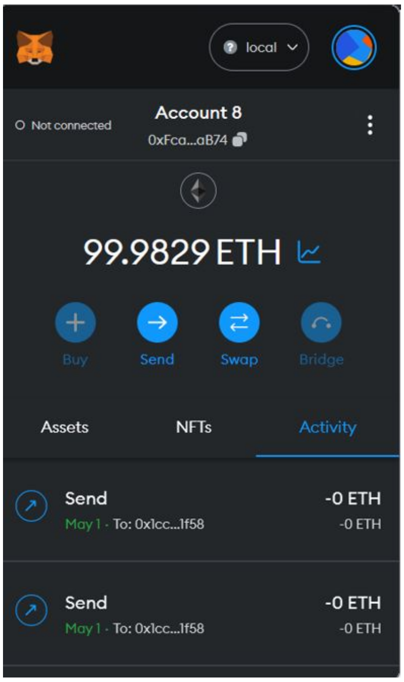
Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network.Since each Ethereum transaction requires computational resources to execute, each transaction requires a fee. Gas refers to the fee required to execute a transaction on Ethereum, regardless of transaction success or failure.

9 User Interface

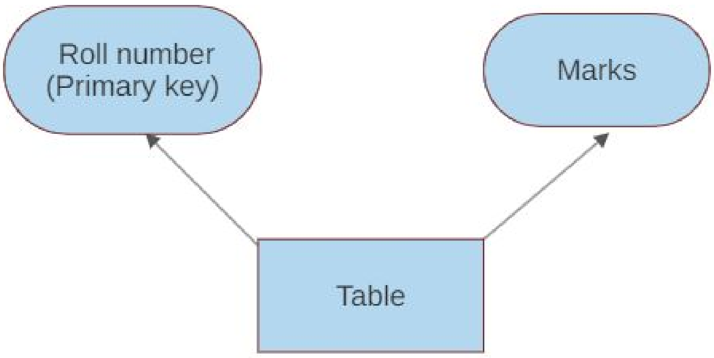
Data of Student Marks

ROLL NO	MARKS
421201	67
421207	78
421202	55
421251	51
421261	99
421262	89
421265	78
421203	57
60	99
421260	99
421263	78
421210	67
421211	98
421213	65
421214	65
421215	45

The JavaScript code uses the Web3.js library to interact with the Ethereum network. It checks if the user has an Ethereum wallet installed and enabled in the browser and connects to the Ethereum network using the Web3 object.Then, it creates an instance of the smart contract using the contract address and ABI, and listens to the form event.

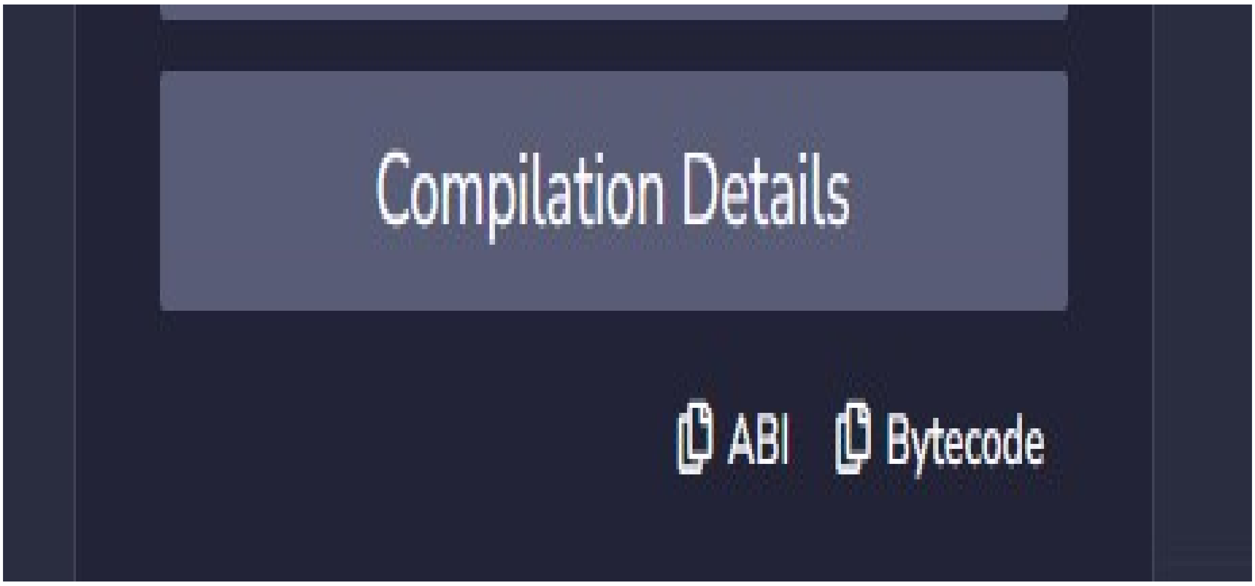


After connecting to Metamask, your web application will be able to interact with the Ethereum blockchain using the accounts that you have authorized within Metamask.Web3.js library send transactions to the Ethereum network. These transactions can involve sending ETH or tokens, executing smart contract functions, or interacting with other applications on the network.



10 Experimental Procedure

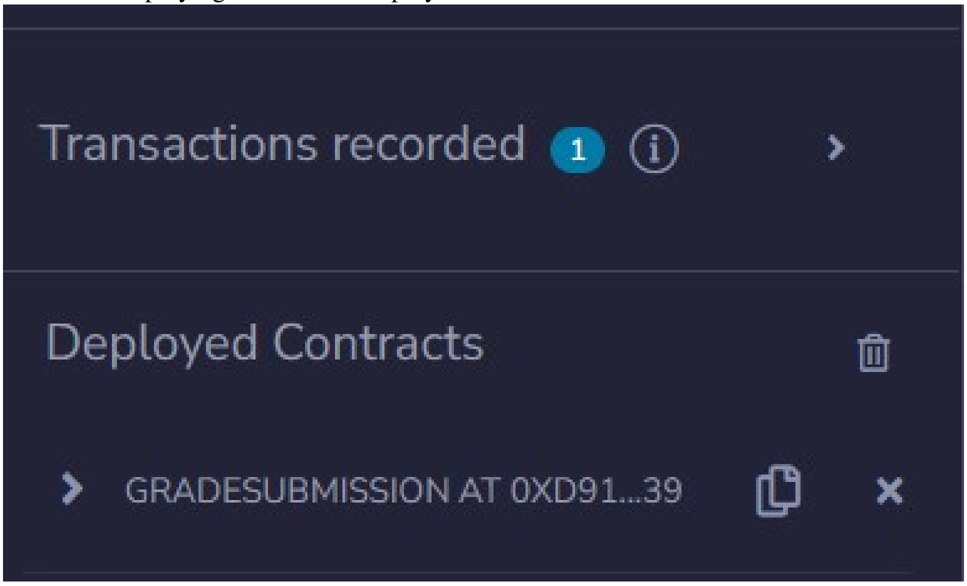
Create a smart contract in Solidity using Remix IDE,open the Remix IDE in web browser. Once the IDE is opened, create a new file and give it a name with a ".sol" extension. Next, start writing smart contract code in the editor provided by Remix IDE.Once you have written your smart contract code, you can compile it using the Solidity compiler provided by Remix IDE.Soon after the compilation you will get an ABI(Abstract Binary Interface code).



Install metamask Legacy web3 extension on Firefox and install Ganache. Take an private key from any of the accounts in Ganache and import them on metamask.

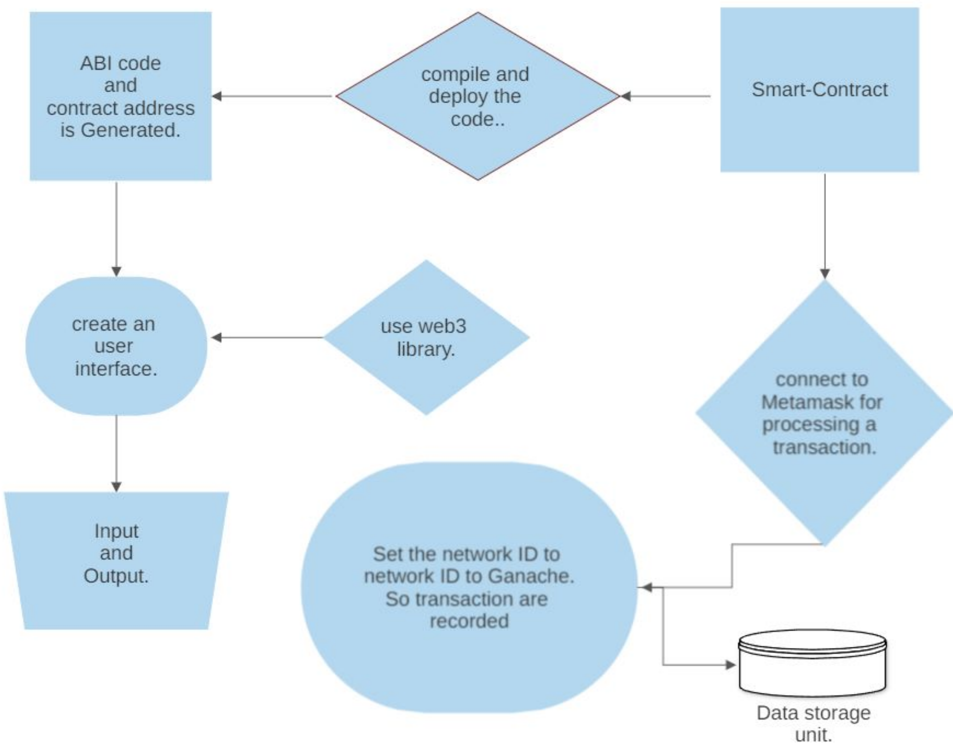
Set the environment to Injected provider metamask in remix IDE and set the network ID to the network ID of the metamask and then deploy the code run Transactions section in remix IDE.

Soon after deploying the code an Deployed Contracts is created.

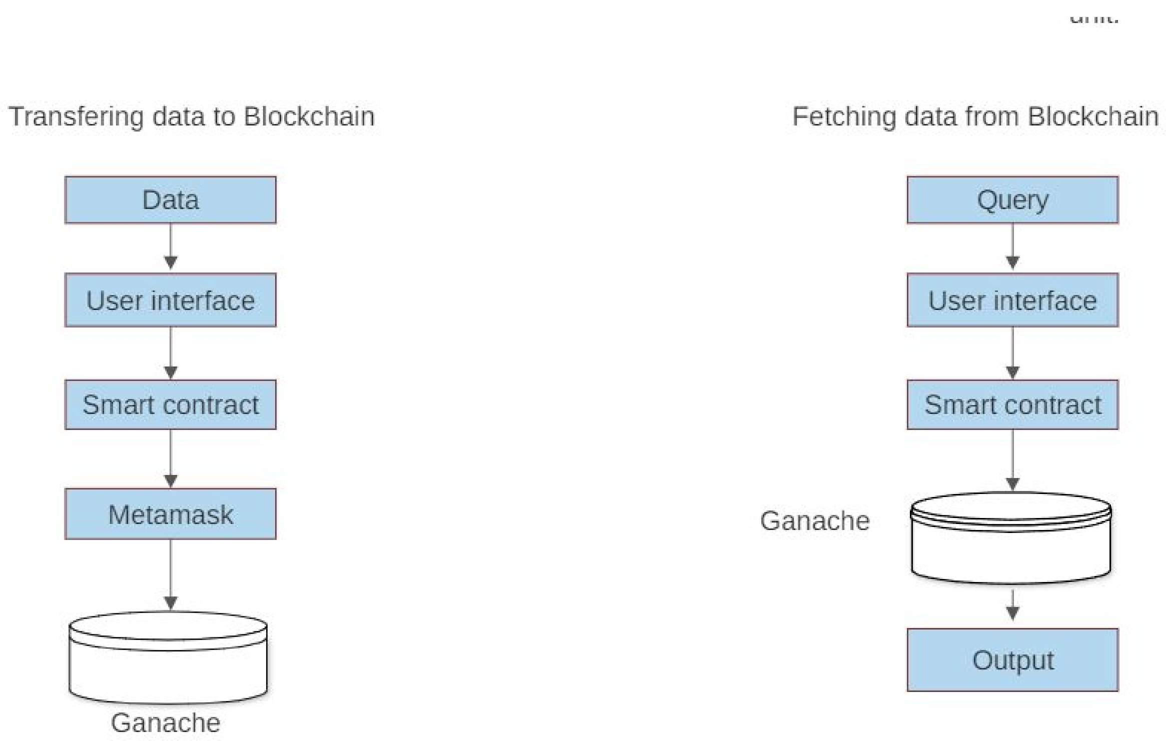


Web3.js is a popular JavaScript library that allows developers to interact with the Ethereum blockchain. It provides a set of APIs that enable you to build decentralized applications (dApps) that interact with smart contracts, send and receive Ether and other tokens, and manage user accounts.Web3.js allows you to interact with smart contracts deployed on the Ethereum network. You can call functions on a contract, read data from it, and listen to events emitted by it.Web3.js makes it easy to create and send transactions to the Ethereum network. You can set the gas limit and gas price for transactions,and monitor their status.Using frontend Interface with web3 we can interact with the Blockchain.

11 Activity Diagram



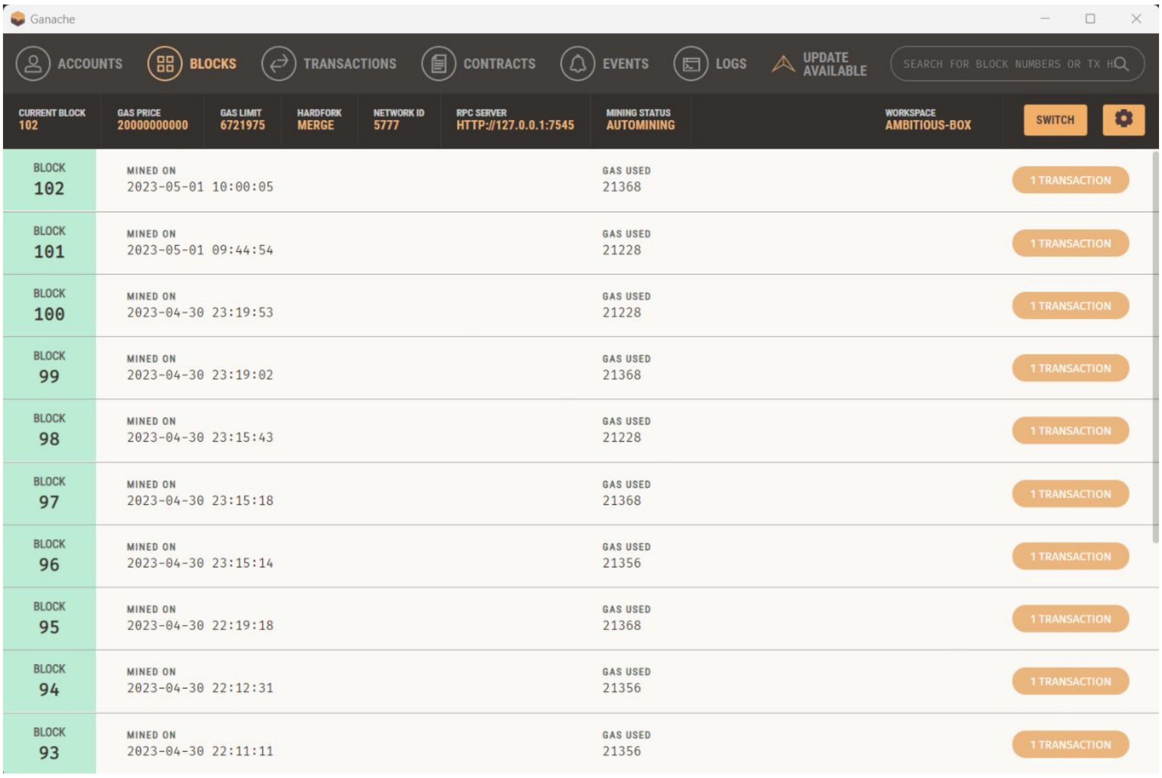
12 Flow of Execution



Any kind of data can be inserted into blockchain. Enter the data in the front-end environment , transfers the data to Ganache as a transaction through Metamask. Every transfering of data is recorded as a transaction and stored in Gabache Blocks along with the time.fetching data from the Ethereum blockchain using MetaMask and Ganache requires setting up a development environment, connecting to the network, writing code, and using Web3.js to interact with the blockchain and smart contracts.

13 Results and Discussions

The ledger framework for storing roll no and marks of students has been successfully developed. The framework is designed to store the roll no and marks of students in a secure and transparent manner. The ledger framework has been developed using blockchain technology, which ensures the security and integrity of the data stored in the ledger.The ledger framework also includes a consensus mechanism, which ensures that all the nodes in the network agree on the validity of the ledger data. This mechanism prevents any malicious activity, such as tampering with the ledger data.



The ledger framework for storing roll no and marks of students is an innovative solution to the problem of secure and transparent record-keeping. The use of blockchain technology ensures that the ledger data is secure, immutable, and tamper-proof. This eliminates the need for a centralized authority to maintain and verify the ledger data, which can be vulnerable to hacking and manipulation.

14 Conclusion

In conclusion, the ledger framework for storing roll no and marks of students provides a secure and transparent solution for educational record-keeping. The use of blockchain technology and a consensus mechanism ensures the integrity and accuracy of the ledger data, while the decentralized nature of the network eliminates the need for a centralized authority to maintain the records. The ledger framework has the potential to improve the efficiency and effectiveness of educational processes, and can be easily integrated with other educational systems. Overall, the ledger framework for storing roll no and marks of students is a promising solution for the challenges faced in educational record-keeping.

15 References

1. :https://github.com/NamChain-Open-Initiative-Research-Lab/Blockchain-Terminologies: Ramaguru R, Minu M. Blockchain Terminologies. NamChain Open InitiativeResearch Lab (2021).
2. :https://doi.org/10.1007/978-981-15-7078-043: Kripa M., Nidhin Mahesh A., Ramaguru R., Amritha P.P. (2021) BlockchainFramework for Social Media DRM Based on Secret Sharing. In: Senjyu T., Ma-halle P.N., Perumal T., Joshi A. (eds) Information and Communication Technologyfor Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies,vol 195. Springer, Singapore.
3. :https://www.amrita.edu/aums/about