CryptoX : BLOCKCHAIN - BASED CRYPTOCURRENCY

MINI PROJECT

Submitted in partial fulfillment of the requirements for the award of the degree

Of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

K.Jyothsna
20331A0582

L.Dhanush
20331A05A0

I.Swathi
20331A0572

N.Suseel Kalyan
20331A05C4

L.Srinivas
20331A0599

Under the Supervision of
Mrs. K. Vindhya Rani

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MVGR COLLEGE OF ENGINEERING
(Autonomous)
VIZIANAGARAM-535005, AP (INDIA)
(Accredited by NBA,
NAAC, and
Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

2022-2023

CERTIFICATE

This is to certify that the project report entitled "**CryptoX: Blockchain Based Cryptocurrency**" being submitted by **K.Jyothsna, I.Swathi, L.Srinivas, L.Dhanush, N.Suseel Kalyan** bearing registered numbers **20331A0582, 20331A0572, 20331A0599, 20331A05A0, 20331A05C4** respectively, in partial fulfillment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2022-2023.

Mrs. K. Vindhya Rani
Assistant Professor
Department of CSE

Dr P. Ravi Kiran Varma
Head of the Department
Department of CSE

ACKNOWLEDGEMENT

We place on record our heartfelt appreciation and gratitude to **Mrs. K. Vindhya Rani** for the immense cooperation and navigation as mentor in bringing out the project work under his guidance. His uncompromising attitude to bring out the best with constructive suggestions inspired us to achieve the results set forth for the project work. We are deeply indebted to him for his excellent, enlightened and enriched guidance.

We consider it our privilege to express our deepest gratitude of **Dr. P. Ravi Kiran Varma,** Head of the Department, for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We also thank **Dr. K. V. L. Raju**, principal for extending his utmost support and cooperation in providing all the provisions for the successful completion of the project. We sincerely thank all the members of the staff in the Department of Computer Science & Engineering for their sustained help in our pursuits. We thank all those who contributed directly or indirectly in successfully carrying out this work.

ABSTRACT

CryptoX is a blockchain-based cryptocurrency that operates independently of a central bank or government. It uses advanced cryptography to secure transactions and control the creation of new units, ensuring decentralization, security, privacy, and global accessibility. The blockchain technology enables secure, transparent, and tamper-proof transactions, making it virtually impossible for anyone to hack or steal funds. CryptoX is created through a process called mining, and once created, it can be bought and sold on cryptocurrency exchanges. As blockchain technology evolves, we can expect to see even more innovative use cases for CryptoX and other cryptocurrencies.

Ultimately, the project aims to contribute to the growing ecosystem of decentralized digital currencies, and to create a currency that has the potential to transform the way we think about money and finance. Additionally, the potential use cases and impact of the new currency on the financialsystem will be explored.

TABLE OF CONTENTS

INTRODUCTION

This project involves building a new cryptocurrency using blockchain technology. The goal is to create a decentralized, secure, and efficient currency that can be used for various purposes such as payments and investments. A user-friendly wallet will also be created to enable users to send and receive the new currency. Scalability and compliance with relevant regulations will be key considerations throughout the project.

The cryptocurrency world may seem daunting to the average investor, especially for those without technical knowledge of blockchain and smart contracts. However, the prospects of many new digital currencies have drawn in all types of investors, including those who might have otherwise been cautious about investing in a digital token or cryptocurrency.

While investors can certainly be successful in the cryptocurrency space without having under-the-hood technical knowledge, a basic understanding of some of the most important properties of many of the current digital currencies is undoubtedly helpful in guiding an investor toward the safest and soundest financial decisions. One of the major concepts that govern a large portion of the space, and which is especially relevant to smart contracts and smart property, is what is known as the ERC20 token standard.

"ERC20" refers to a scripting standard used within the Ethereum blockchain. This technical standard dictates several rules and actions that an Ethereum token or smart contract must follow and steps to be able to implement it. It is perhaps easiest to think of ERC20 as a set of basic guidelines and functions that any new token created in the Ethereum network must follow.

OBJECTIVES:

- **Decentralization:** One of the primary goals of cryptocurrencies is to create a decentralized system that is not controlled by any central authority, such as a government or a bank. By building your own cryptocurrency,you can create a platform that is not subject to the same regulations and limitations as traditional financial systems.
- **Transparency:** Cryptocurrencies are built on blockchain technology, which allows for transparent andsecure transactions. By creating your own cryptocurrency, you can ensure that all transactions are record on a public ledger, providing greater transparency and accountability.
- **Faster transactions:** Cryptocurrencies can offer faster transaction speeds than traditional financial systems, which can be particularly useful for international transactions.
- **Privacy:** Cryptocurrencies can offer a high degree of privacy and anonymity, which can be important forusers who want to protect their financial information.
- **Global:** CryptoX is a global currency that can be used anywhere in the world, making it a convenient option for international transactions. It eliminates the need for currency exchange and reduces the fees associated with international transactions.

Problem Statement:

The current financial system is centralized, with governments and financial institutions having significant control over the flow of money. This can lead to issues such as censorship, high transaction fees, and slow processing times. Additionally, traditional currencies are not alwaysaccessible to everyone, particularly those in underbanked or unbanked communities. Cryptocurrencies have the potential to offer a decentralized, secure, and efficient alternative totraditional currencies. This project aims to address these challenges by building a newcryptocurrency using blockchain technology.

Scope:

- CryptoX has a broad scope, as it can be used as a digital currency for a wide range of transactions.
- It can be used to buy goods and services, make international payments, and even as a store of value.
- The blockchain technology that CryptoX is built on has the potential to be used for a variety of other applications beyond just digital currency, such as smart contracts and decentralized applications.

Limitations:

- One of the main limitations of CryptoX is its volatility. The value of CryptoX can fluctuate rapidly, making it difficult to use as a stable store of value. Additionally, while CryptoX offers a high degree of privacy and anonymity, it has been criticized for its use in illegal activities such as money laundering and illicit drug trades.
- CryptoX operates independently of any government or financial institution, it is not backed by any physical assets, which may limit its adoption in certain industries and regions.
- The mining process used to create new CryptoX units requires significant computational resources, which may limit its scalability and accessibility to a wider audience.

LITERATURE REVIEW

- https://www.researchgate.net/publication/337152829_Blockchain_and_Cryptocurrencies_Technology_a_survey
- https://www.mdpi.com/2504-2289/2/4/34
- https://www.ijirt.org/Article?manuscript=153630
- https://www.jetir.org/view?paper=JETIRZ006066
- https://jfin-swufe.springeropen.com/articles/10.1186/s40854-021-00321-6

Overall, the literature review highlights the importance of blockchain-based cryptocurrencies in various industries and their potential to disrupt traditional systems. However, the review also highlights several limitations, such as scalability and energy consumption, that need to be addressed to fully realize the potential of blockchain-based cryptocurrencies. Future research in this area should focus on developing innovative solutions to address these limitations and make blockchain-based cryptocurrencies more widely adopted.

Algorithms used :

❖ Consensus algorithm

Consensus algorithm is a method used to achieve agreement, trust and security across a decentralized network. In the context of blockchain and cryptocurrencies proof-of-work (POW) and proof-of-stake(POS) are the most prevalent consensus algorithms. This mechanism plays an essential part of securing information by encrypting it and using automated group verification.

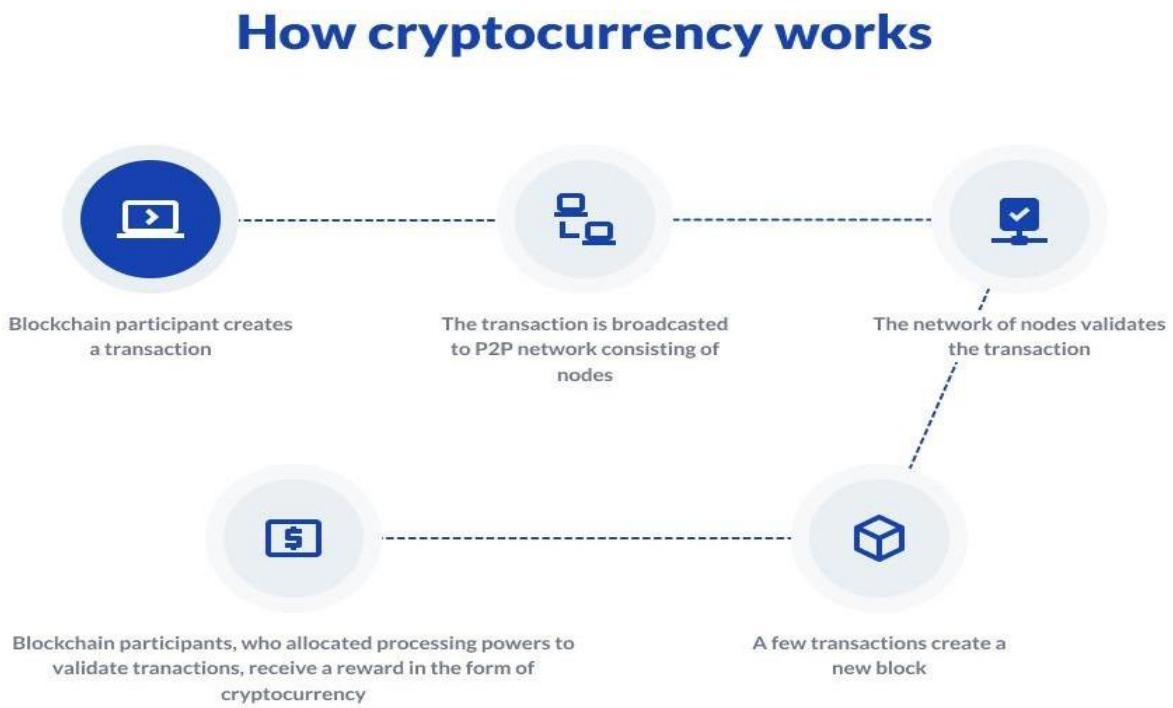
❖ Proof of work

Proof of work is a common consensus algorithm used by the most popular cryptocurrency Networks like Bitcoin. It requires a participant node to prove that the work done and submitted By them qualifies them to receive the right to add new transactions to the blockchain.

Techniques Used:

- **Privacy techniques:** Privacy is an important concern in blockchain-based cryptocurrencies, and various techniques have been used to enhance privacy, such as zero-knowledge proofs, ring signatures, and stealth addresses.
- **Smart contracts:** Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. Smart contracts have been used in blockchain-based cryptocurrencies to automate various processes, such as asset transfers, identity verification, and voting.
- **Blockchain interoperability:** Interoperability is the ability of different blockchain networks to communicate and work together seamlessly. Several approaches have been proposed to achieve blockchain interoperability, such as sidechains, cross-chain atomic swaps, and blockchain bridges.
- **Cryptographic techniques:** Cryptography plays a crucial role in securing blockchain-based cryptocurrencies. Various cryptographic techniques have been used in related work, such as hash functions, public-key cryptography, and digital signatures.

PROBLEM DEFNITION



1. **Nodes:** CryptoX is based on a decentralized network of nodes, which are computers or servers that host a copy of the blockchain. Each node maintains a copy of the entire blockchain and verifies transactions.
2. **Transactions:** Users can send and receive CryptoX tokens by creating transactions on the network. Transactions contain information about the sender, the recipient, and the amount of CryptoX being transferred.
3. **Validation:** Transactions are validated by the network's consensus algorithm, which ensures that the transaction is valid and that the sender has sufficient funds to make the transfer.
4. **Mining:** Once a transaction is validated, it is added to a block, which is then added to the blockchain. Miners, who are nodes that participate in the validation process, compete to create new blocks by solving a complex mathematical puzzle. The first miner to solve the puzzle and create a new block is rewarded with a certain number of CryptoX tokens.
5. **Smart Contracts:** Smart contracts are self-executing programs that run on the blockchain. They can be used to automate various processes, such as asset transfers, identity verification, and voting. Smart contracts can be written in various programming languages,

such as Solidity, and deployed on the blockchain.

6. **Cryptography:** Cryptography plays a crucial role in securing CryptoX transactions. Transactions are secured using public-key cryptography, which involves using a pair of keys - a public key and a private key - to encrypt and decrypt information. Transactions are also hashed using a cryptographic hash function, which generates a unique digital fingerprint of the transaction that can be used to verify its integrity.

The Cryptocurrency Mining

Cryptocurrency mining creates a question on its legalization when hackers try to mine Cryptocurrency just like the gold mining. The figure above tries to correlate the real world mining and the digital mining.

HARDWARE/SOFTWARE PROCUREMENTS

1. REMIX- ETHEREUM IDE

REMIX is an integrated development environment (IDE) for writing, testing, and deploying smart contracts on the Ethereum blockchain. It is a web-based IDE that allows developers to write and debug smart contracts in various programming languages, such as Solidity and Vyper.

Here are some key features and functionalities of REMIX:

- **Code Editor:** REMIX provides a code editor with syntax highlighting and auto-completion for writing smart contracts in Solidity or Vyper. It also provides a built-in compiler that can compile the code and generate bytecode.
- **Debugger:** REMIX has a built-in debugger that allows developers to step through their code line by line, set breakpoints, and inspect variables. This helps developers identify and fix errors in their smart contracts.
- **Deploy and Run Contracts:** Developers can use REMIX to deploy and run their smart contracts on the Ethereum blockchain. REMIX provides a user-friendly interface for deploying contracts and interacting with them.

2. METAMASK

Metamask is a popular browser extension and mobile app that allows users to interact with the Ethereum blockchain and its decentralized applications (dApps). It serves as a digital wallet, providing users with a secure and convenient way to manage their Ethereum assets and interact with the decentralized web.

3. SOLIDITY PROGRAMMING LANGUAGE

Solidity is a high-level programming language that is used for writing smart contracts on the Ethereum blockchain. It is an object-oriented language that is influenced by C++, Python, and JavaScript. Solidity is the most widely used language for developing Ethereum-based applications and smart contracts.

HARDWARE REQUIREMENTS

- **Device:** Laptop, Smart Phones or Desktop Computer
- **Processor:** CORE i3 (3rd Gen minimum) and above
- **RAM:** 4GB (minimum) and above
- **Hard disk:** 100 GB (minimum) and above

IMPLEMENTATION

Safe Math Interface

SafeMath is a library in the Solidity programming language that provides arithmetic operations with overflow and underflow protections. The library is designed to prevent common vulnerabilities that can occur in smart contracts, such as integer overflow and underflow.

The SafeMath library provides functions for common arithmetic operations, such as addition, subtraction, multiplication, and division. These functions ensure that the result of the operation does not exceed the maximum or minimum value that can be stored in the data type.

Code Snippet :

```
contract SafeMath {

    function safeAdd(uint a, uint b) public pure returns (uint c) {
        c = a + b;
        require(c >= a);
    }

    function safeSub(uint a, uint b) public pure returns (uint c) {
        require(b <= a);
        c = a - b;
    }

    function safeMul(uint a, uint b) public pure returns (uint c) {
        c = a * b;
        require(a == 0 || c / a == b);
    }

    function safeDiv(uint a, uint b) public pure returns (uint c) {
        require(b > 0);
        c = a / b;
    }
}
```

This is a library contract called SafeMath that defines functions for performing arithmetic operations safely, to avoid integer overflow and underflow issues.

Token Contract:

This is the main contract QKCToken that implements the ERC20 token standard and uses the SafeMath library. It declares the symbol, name, decimals, _totalSupply, balances, and allowed state variables.The constructor initializes the token properties by setting the symbol, name, decimals, _totalSupply and assigning the entire _totalSupply to the contract owner's address, which is a common practice for creating a fixed supply token. The token contract implements the ERC20 standard functions by providing implementations for each of them.

Code Snippet:

```
contract QKCToken is ERC20Interface, SafeMath {
    string public symbol;
    string public  name;
    uint8 public decimals;
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;



    function totalSupply() public constant returns (uint) {
        return _totalSupply  - balances[address(0)];
    }


    function transfer(address to, uint tokens) public returns (bool success) {
        balances[msg.sender] = safeSub(balances[msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(msg.sender, to, tokens);
        return true;
    }

    function approve(address spender, uint tokens) public returns (bool success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        return true;
    }
```

- The constructor initializes the token properties by setting the symbol, name, decimals, _totalSupply and assigning the entire _totalSupply to the contract owner's address, which is a common practice for creating a fixed supply token.
- The transfer() function is used to transfer tokens from the sender's account to a specified recipient's account.

- The approve() function is used to approve a spender to transfer tokens from the owner's account.

```
    function transferFrom(address from, address to, uint tokens) public returns (bool success) {
        balances[from] = safeSub(balances[from], tokens);
        allowed[from][msg.sender] = safeSub(allowed[from][msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(from, to, tokens);
        return true;
    }

    function allowance(address tokenOwner, address spender) public constant returns (uint remaining) {
        return allowed[tokenOwner][spender];
    }

    function approveAndCall(address spender, uint tokens, bytes data) public returns (bool success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        ApproveAndCallFallBack(spender).receiveApproval(msg.sender, tokens, this, data);
        return true;
    }

    function () public payable {
        revert();
    }
}
```

- The transferFrom() function is used to transfer tokens from one address to another address, on behalf of the owner.
- The allowance() function is used to check the amount of tokens that a spender is allowed to spend on behalf of the owner.
- The approveAndCall() function is used for approving and calling a function in a single transaction.
- The fallback function is declared as revert(), which means any incoming ether transactions will be rejected and the transaction will revert.

RESULT

After compiling the code in the Remix IDE, the next step is to deploy the contract. Upon doing so, the Metamask dialog box will appear, prompting the user to confirm their password and the transaction amount of gas fee.

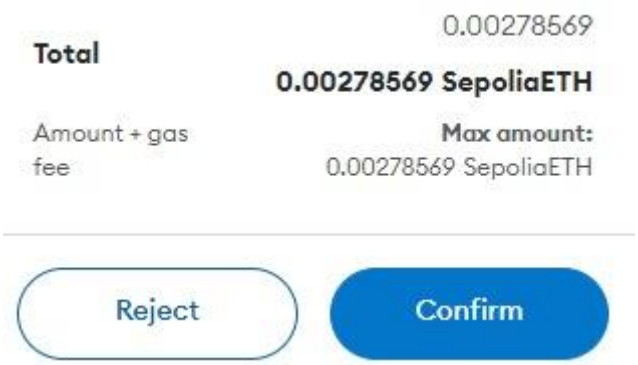


Fig: Shows the prompting the user to confirm their password

Once the transaction has been confirmed, a notification will be displayed indicating that the transaction has been successfully processed.

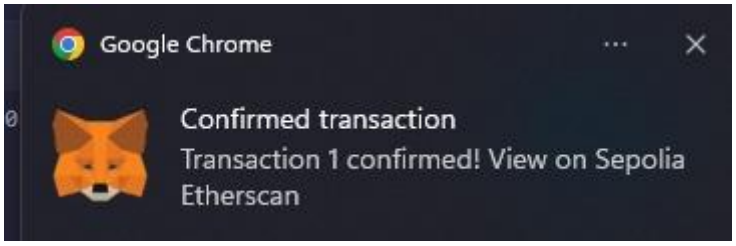


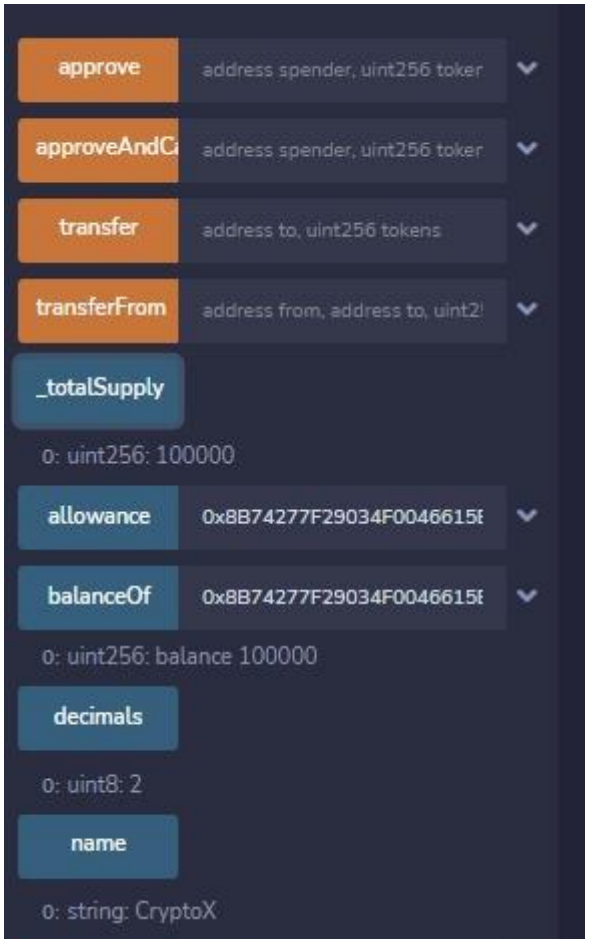
Fig: Shows the user that transaction is completed

The subsequent step involves accessing the various functions available to input the public keys and verify whether the expected values have been achieved.

The functions involved in this process are:

TotalSupply: provides information about the total token supply

BalanceOf: provides account balance of the owner's account



Transfer: executes transfers of a specified number of tokens to a specified address

TransferFrom: executes transfers of a specified number of tokens from a specified address

Approve: allow a spender to withdraw a set number of tokens from a specified account

Allowance: returns a set number of tokens from a spender to the owner

Fig: Shows the various functions available to input the public keys

Once the deployment process has been completed successfully, the user can view the transaction details on Ether scan. These details include the transaction hash, contract address, from address, timestamp, transaction fee, and gas price.

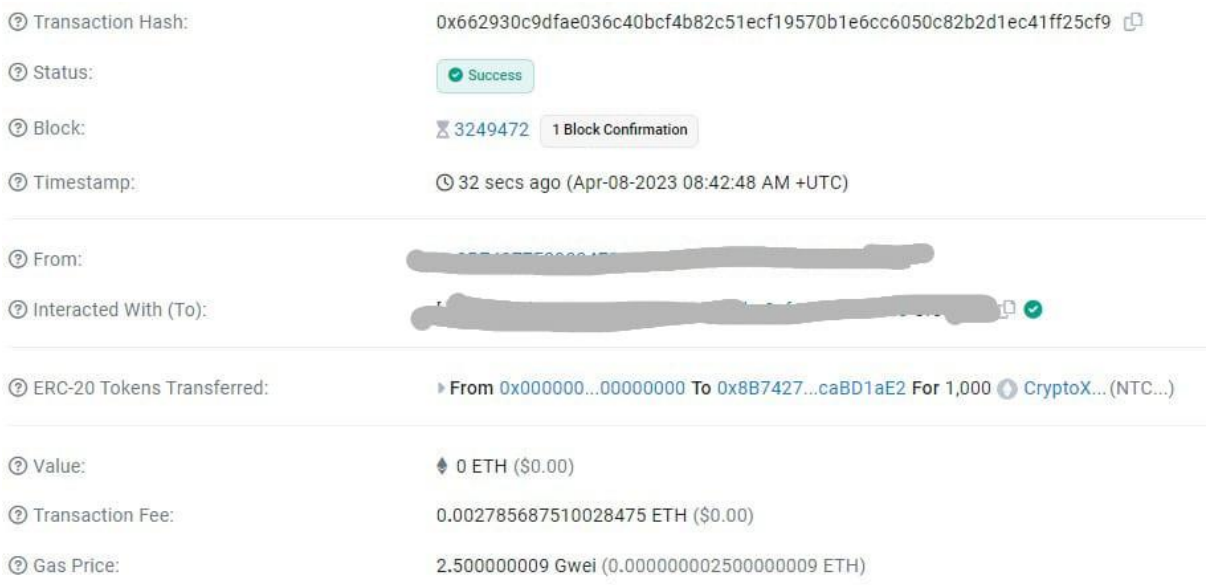


Fig: Shows the user the transaction details on Ether scan

Finally, the deployment of the contract will be reflected in the Metamask wallet.

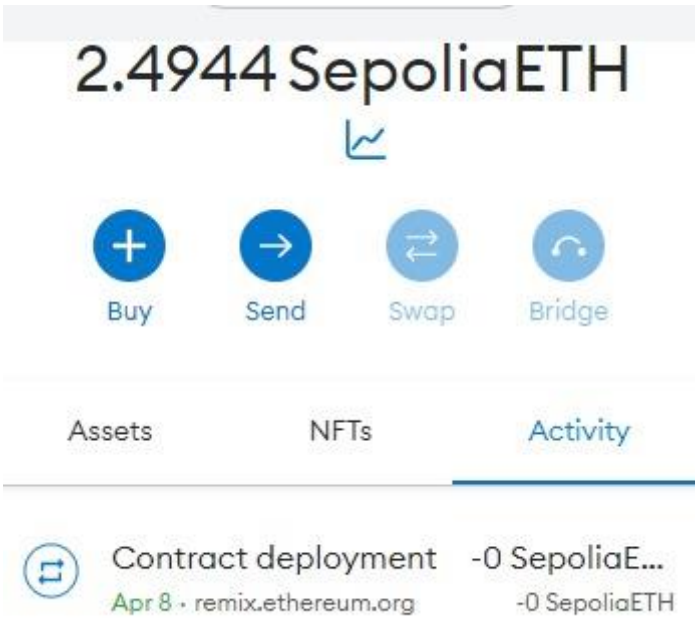


Fig: Shows the contract is deployed

CONCLUSION

In conclusion, the use of blockchain technology and cryptocurrencies is rapidly growing and has the potential to revolutionize various industries. The CryptoX project is a blockchain-based cryptocurrency that is designed to provide fast and secure transactions with low fees.

The project uses the Ethereum blockchain, which allows for the development of smart contracts using the Solidity programming language. The use of smart contracts enables the implementation of complex logic and automation, providing transparency and security to transactions.

The development of the CryptoX project requires a range of methodologies, algorithms, and techniques. These include the use of the Remix Ethereum IDE for writing, testing, and debugging smart contracts, the MetaMask wallet for interacting with the Ethereum network, and the SafeMath library for preventing vulnerabilities in arithmetic operations.

Overall, the CryptoX project is an example of the potential of blockchain technology and cryptocurrencies to transform the way we conduct transactions and interact with financial systems. With ongoing development and innovation in the blockchain space, we can expect to see even more exciting applications of this technology soon.

OUTCOMES

1. Decentralization can be obtained as we have connected the smart contract to the Sepolia Network which is a test network of blockchain.
2. Transparency is achieved through CryptoX as all the transactions are recorded on a public ledger protected by a secure hash which allows secure transactions.
3. The speed of the transaction is relatively high and sometimes based on the test network. During network congestion the miners prioritize high gas fee transactions rather than less fee contracts.
4. Privacy can be easily obtained through the currency created as every transaction will be stored with a hash on the block.
5. The coin is global but for now it is not publicly available as it may require actual ethers rather than test ethers and coin should be validated on the main network.

Mapping of project objectives (POBJS) to Project Outcome's (POS) with levels of correlation:

POBJS	PO-1	PO-2	PO-3	PO-4	PO-5
POBJ-1	3	2	2	3	1
POBJ-2	2	2	2	2	2
POBJ-3	3	3	2	2	2
POBJ-4	2	2	2	3	1
POBJ-5	2	2	2	2	1

REFERENCES

- https://ethereum.org/en/developers/docs/standards/tokens/erc-20/
- https://www.investopedia.com/tech/why-crypto-users-need-know-about-erc20-token-standard/#:~:text=An%20ERC20%20token%20is%20a,standard%20was%20implemented%20in%202015
- https://www.jetir.org/view?paper=JETIRZ006066https://github.com/nathan149/CustomCryptocurrency/blob/master/gymcoin/blockchain.py
- https://www.toptal.com/ethereum/create-erc20-token-tutorial
- https://github.com/jspruance/erc20-tutorial-block-explorer
- https://medium.com/coinmonks/create-your-own-cryptocurrency-in-ethereum-blockchain-40865db8a29f