A

PROJECT REPORT

ON

Crypto Blockchain Transaction

Submitted by

Avlani Hardish (19IT465)

Sanket Detroja (19IT413)

Smit Bhansali (19IT467)

Prince Sheth (19IT468)

For Partial Fulfilment of the Requirements for Bachelor of Technology in Information Technology

Guided by

Prof. Vishal Polara

December, 2022

Information Technology Department
Birla Vishvakarma Mahavidyalaya Engineering College
(An Autonomous Institution)
Vallabh Vidyanagar – 388120
Gujarat, INDIA

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2022-23, Semester VII

CERTIFICATE

This is to certify that the project work entitled **Crypto Transaction Using Blockchain** has been successfully carried out by **19IT468- Prince Sheth, 19IT467- Smit Bhansali, 19IT465- Hardish Avlani, 19IT413- Sanket Detroja** for the subject **Project I (4IT31)** during the academic year 2022-23, Semester- VII for the partial fulfilment of Bachelor of Technology in Information Technology. The work carried out during the semester is satisfactory.

Prof. Vishal Polara
Project Guide,
IT Department
BVM

Dr. Keyur Brahmbhatt
Head of Department,
IT Department
BVM

ACKNOWLEDGEMENT

We have designated my entire time and efforts in this project along with its research. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them for theirvaluable Assistance.

We are extremely grateful to my project guide, **Prof. Vishal Polara**, faculty of Information Technology, for guiding us throughout the project and for the effective doubt solving sessions with patience and knowledge.

We are grateful to our Course Co-ordinator **Dr. Zankhana Shah** for giving us the support and encouragement that was necessary for the completion of this project.

We would like to express my gratitude to the HOD of IT Department **Dr. Keyur Brahmbhatt** and we are also grateful to all our **faculty members of BVM Engineering College**, IT Department for their kind cooperation and encouragement which helped me in completing this project and preparing the report.

Last but not the least, we would also like to thank my colleagues, who have co – operated during the preparation of my report and without them this project has not been possible. Their ideas helped me a lot to improve my project report.

ABSTRACT

Blockchain is an element of technology used most notably in Cryptocurrency trading. However, studies and substantial experiments are underway for its application in various financial transactions. In recent times, Blockchain has received extensive attention in digital transaction and trading, and the top of that, it provides more security and transparency of transaction records.

This chapter expounds on the main concepts of Blockchain technology and its cutting-edge applications. With the help of Blockchain technology, we can make transactions in a decentralized manner with no involvement of any third-party system. It runs as smoothly as other centralized applications with the advantage of blockchain qualities.

In this project, we have utilized Web 3.0, the latest internet technology that is helpful in Blockchain to achieve real-world human communication, and for transactions, we have used Solidity. First, we have to connect our application to the MetaMask Wallet. Then we can send Ethereum from one account to another using the hexadecimal address of the wallet. Here, we also have to send a message and a unique keyword to generate a GIF for the transaction. After that, these transactions can be seen in the transactions part of the page with the individual GIF, the receiver's address, message, date and time, etc.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Aim of the project

Modern real world application of block chain technology. Web application connected to the Metamask pairing interaction with smart contracts sending Ethereum through the Blockchain networking. Creating web application and pair it to your Ethereum wallet using Metamask.

This will allow users to send transaction through the Blockchain technology. Each transaction paired with a GIF and it will be forever stored on Blockchain. It will send crypto across the world.

1.2 Project Scope

Transparent transaction using Blockchain technology increases the security of the transaction of crypto that races towards the cybersecurity.

All the data is secure and verified. The encryption is done through cryptography to eliminate vulnerabilities such as unauthorized data tampering and this will increase the crypto payments.

1.3 Project Objectives

- It will enable users to perform a quick Ethereum transaction.
- It is a highly secure, quick, and reliable transaction platform.
- It will ensure a security method for the transaction will not be compromised.
- It provides all necessities like news, market, converter, and FAQ to perform a safe transaction.

1.4 Project Modules

1. Registration
2. MetaMask connection
3. Transaction of crypto
4. Creating gif

1.5 Project Basic Requirements

1.5.1 Hardware Requirements

- Hardware that supports web app (E.g., Mobile, Computer)

1.5.2 Software Requirements

- Front-End: React-js
- Back-End: Solidity
- Others: Domain name, Hosting, sever and cloud storage

1.5.3 Software requirements for our clients

- Windows 7 or higher OS
- Google chrome or any other safe browser

Chapter 2: Literature review

2.1 Introduction

This work provides a systematic literature review of Blockchain-bases application across multiple domains. The aim is to investigate the current state of Blockchain technology and its application and highlight how specific characteristics of this disruptive technology can revolutionize "business-as-usual" practices.

In principle, a Blockchain should be considered as a distributed opened-only timestamped data structure. Blockchain allows us to have a distributed peer-to-peer network where non-trusting members can verifiably interact with each without the need for a trusted authority (K. Christidis, M. Devetsikiotis: Blockchains and smart contracts for the internet of things (2016)). To achieve this one can consider Blockchain as a set of interconnected mechanisms which provide specific features to the infrastructure, as illustrated in below figure 1.

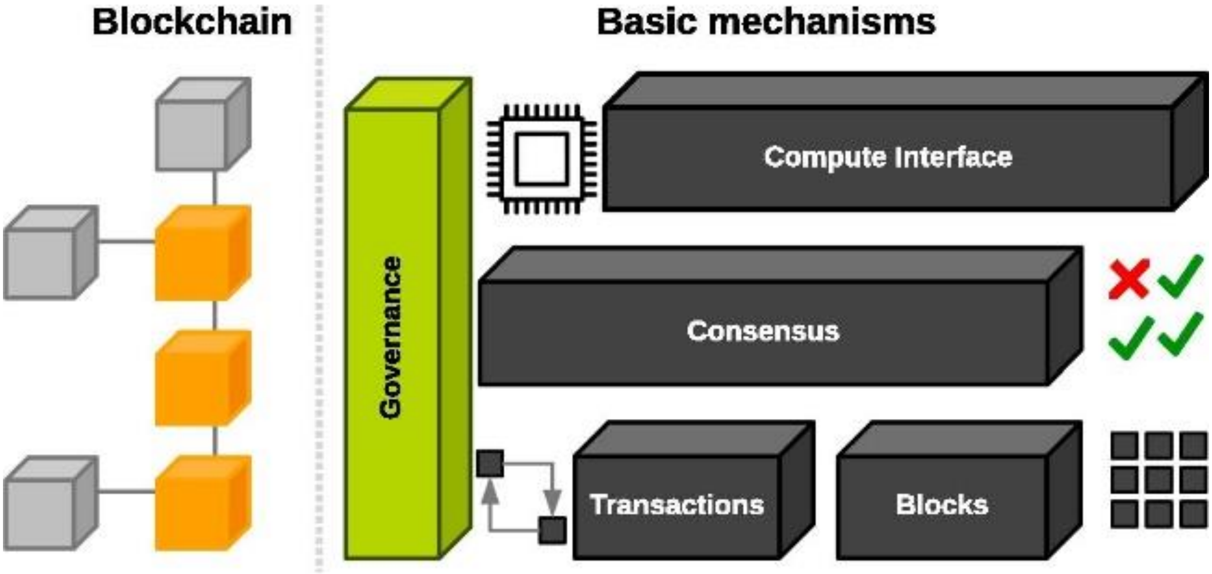


Figure 1 Blockchain Architecture

At the lowest level of this infrastructure, we have the signed transactions between peers. These transactions denote an agreement between two participants, which may involve the transfer of physical or digital assets, the completion of a task, etc. At least one participant signs this transaction, and it is disseminated to its neighbors. Typically, any entity which connects to the Blockchain is called a node. However, nodes that verify all the Blockchain rules are called full nodes. These nodes group the transactions into blocks and they are responsible to determine whether the transactions are valid, and should be kept in the Blockchain, and which are not. A valid transaction means, for instance, that Bob received one Ethereum from Alice. However, Alice may have tried to transfer the same Ethereum, as it is a digital asset, to Carol. Therefore, nodes must reach to an agreement on which transactions must be kept in the Blockchain to guarantee that there will be no corrupt branches and divergences (M. Vukolić: The quest for scalable Blockchain fabric: proof-of-work vs. BFT replication: International Workshop on Open Problems in Network Security, Springer (2015)).

This is actually the goal of the second Consensus layer. Depending on the Blockchain type, different Consensus mechanisms exist (Mingxiao et al., 2017). The most well-known is the Proof-of-work (PoW). PoW requires solving a complicated computational process, like finding hashes with specific patterns, e.g., a leading number of zeroes (Antonopoulos, 2014), to ensure authentication and verifiability. Instead of splitting blocks across proportionally to the relative hash rates of miners (i.e., their mining power), Proof-of-Stake (PoS) protocols split stake blocks proportionally to the current wealth of miners (Pilkington, 2016). This way, the selection is fairer and prevents the wealthiest participant from dominating the network. Many Blockchains, such as Ethereum (Dannen, 2017), are gradually shifting to PoS due to the significant decrease in power consumption and improved scalability. Other consensus approaches include Byzantine Fault Tolerance (BFT) (Castro and Liskov, 2002) and its variants (Zheng et al., 2016).

An additional layer, the Compute Interface, allows Block chains to offer more functionality. Practically, a Blockchain stores a state which consists e.g., of all the transactions that have been made by the users, thereby allowing the calculation of each user's balance. However, for more advanced applications we need to store complex states which are updated dynamically using distributed computing, e.g., states that shift from one to another once specific criteria are met. This requirement has given rise to SCs which use nodes of the Blockchain to execute the terms of a contract.

Finally, the Governance layer extends the Blockchain architecture to cover the human interactions taking place in the physical world. Indeed, although Blockchain's protocols are well defined, they are also affected by inputs from diverse groups of people who integrate new methods, improve the Blockchain protocols and patch the system. While these parts are necessary for the growth of each Blockchain, they constitute off-chain social processes. Therefore, Blockchain governance deals with how these diverse actors come together to produce, maintain, or change the inputs that make up a Blockchain.1

Current literature categories Blockchain networks in several ways (Buterin, 2015, Zheng et al., 2016, Eris Industries, 2016, Christidis and Devetsikiotis, 2016, Kravchenko, 2016, Wood, 2016). These categories are formed according to the network's management and permissions as public, private and federated. In public Blockchains (permissionless) anyone can join as a new user or node miner. Moreover, all participants can perform operations such as transactions or contracts. In private Blockchains; which along with the federated belong to the permissioned Blockchain category, usually, a whitelist of allowed users is defined with particular characteristics and permissions over the network operations. Since the risk of Sybil attacks is almost negligible there (Swanson, 2015), private Blockchain networks can avoid expensive PoW mechanisms. Instead, a wider range of consensus protocols based on disincentives could be adopted. A federated Blockchain is a hybrid combination of public and private Blockchains (Buterin, 2015, Zheng et al., 2016). Although it shares similar scalability and privacy protection level with private Blockchain, their main difference is that a set of nodes, named leader nodes, is selected instead of a single entity to verify the transaction processes. This enables a partially decentralized design where leader nodes can grant permissions to other users. In this article, we provide a more fine-grained Blockchain network classification than current the state-of-the-art (Buterin, 2015, Zheng et al., 2016, Christidis and Devetsikiotis, 2016, Kravchenko, 2016) because, in addition to classical features such as the ownership and management of the information shared in the Blockchain, we consider features such as transaction approval time, or security aspects such as anonymity. Table 1 summarizes the main characteristics of each Blockchain network regarding efficiency, security and consensus mechanisms.

PROPERTY	PUBLIC	PRIVATE	FEDERATED
CONSENSUS MECHANISM	• Costly PoW • All miners	• Light PoW • Centralised organisation	• Light PoW • Leader node set
IDENTITY	• (Pseudo) Anonymous	• Identified users	• Identified users

ANONYMITY	• Malicious?	• Trusted	• Trusted
PROTOCOL EFFICIENCY &	• Low efficiency	• High efficiency	• High efficiency
CONSUMPTION	• High energy	• Low energy	• Low energy
IMMUTABILITY	• Almost impossible	• Collusion attacks	• Collusion attacks
OWNERSHIP &	• Public	• Centralised	• Semi-Centralised
MANAGEMENT	• Permission less	• Permissioned whitelist	• Permissioned nodes
TRANSACTION APPROVAL	• Order of minutes	• Order of milliseconds	• Order of milliseconds

Table 1 Characteristics of Blockchain

Well-known implementations of public Blockchains include Bitcoin, Ethereum, Litecoin and, in general, most cryptocurrencies (Nakamoto, 2008, Haferkorn and Quintana Diaz, 2015). One of their main advantages is the lack of infrastructure costs: the network is self-sustained and capable of maintaining itself, drastically reducing management overheads. In private Blockchains, the main applications are database management, auditing and, in general, performance demanding solutions (Zheng et al., 2016). Multichain (Greenspan, 2015b) is an example of an open platform for building and deploying private Blockchains. Finally, federated Blockchains are mostly used in the banking and industry sectors (R3, 2015). This is the case of the hyper ledger project (Hyper ledger Project, 2015) which develops cross-industry permission-based Blockchain frameworks. Recently, Ethereum has also provided tools for building federated Blockchains. Other projects such as Cardano (2018) are rather ambitious trying to provide more functionality. For more on Blockchain categorization, the interested reader may refer to Walport, 2016, Swanson, 2015.

Blockchain-based applications:

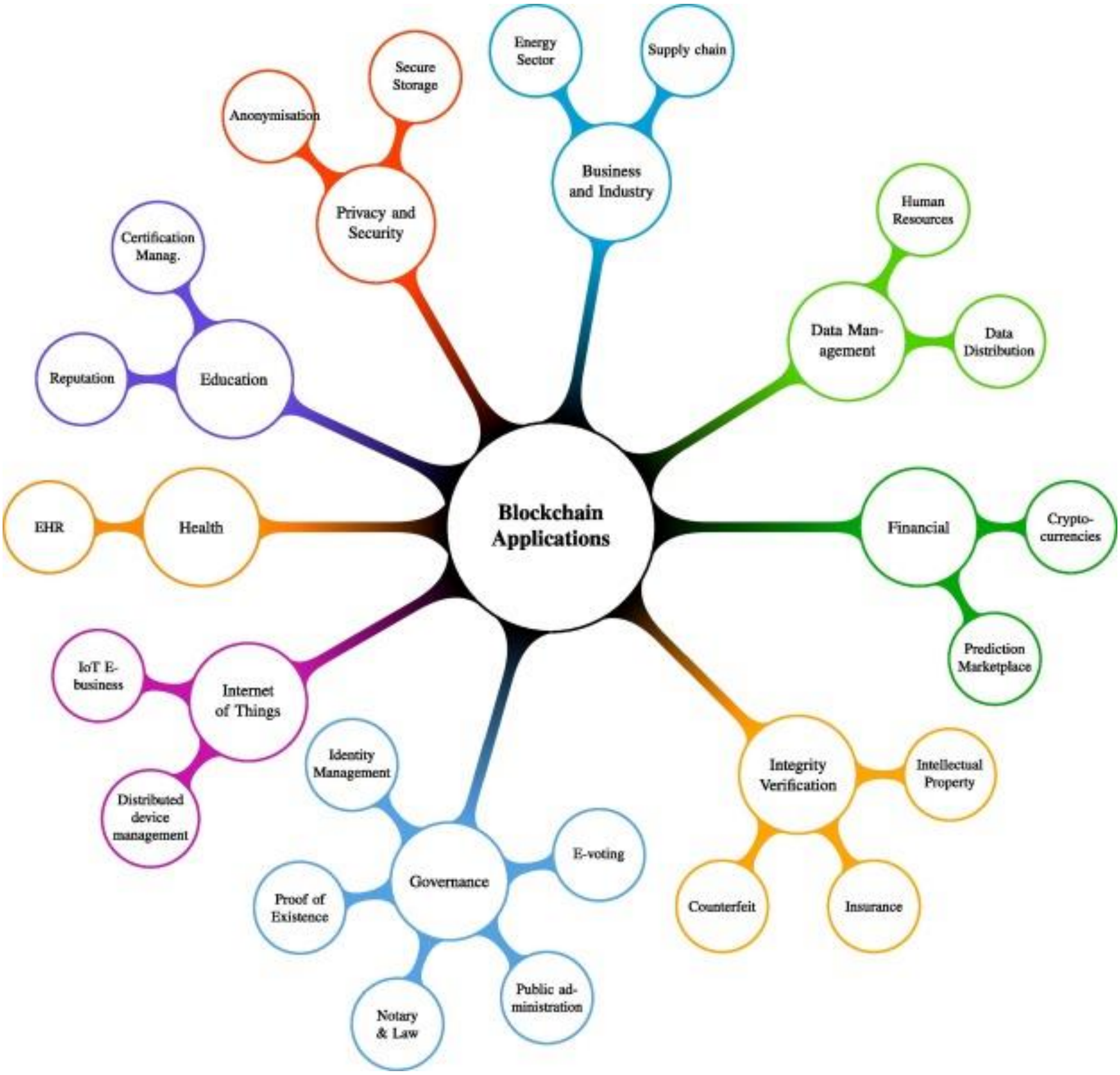


Figure 2 Application of Blockchain

Chapter 3: Analysis, Design Methodology and Implementation Strategy

3.1 Project Feasibility study

We presently portray a normal client interaction with the proposed conspire based on our current framework usage. We make a web application. We use 3.0 in this project and for backend of our project with solidity.

So basically, user have to connect meta-mask with their account. After successfully connect to meta-mask the user to fill the form. In that form user have to enter the address, amount, message and GIF. If the user click on send button, it will take half minute. After that user can check their transaction and its details. In details user can see time and date, address of account in which they paid the Ethereum, message, GIF etc. user can done more and more transactions with more and more security and transparency.

3.2 Detailed Module Description

3.2.1 MetaMask Connection

- Creation of wallet in MetaMask
- Connection of wallet in MetaMask
- Disconnect MetaMask wallet

3.2.2 Transaction of Crypto

- Transaction stored in Blockchain using solidity.
- Contracts for every event like addToBlockchain(), getAllTransactions().

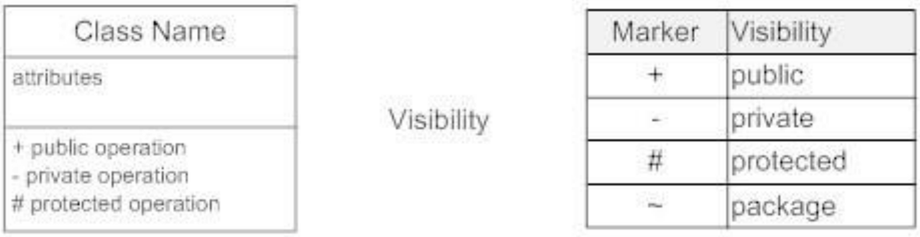
3.2.3 Creating GIF

- Permanent Unique creation of gif that reflects' transaction ID.

3.3 Project SRS

3.3.1 Class Diagram:

Visibility: Use visibility markers to signify who can access the information contained within a class.



Marker	Visibility
+	public
-	private
#	protected
~	package

Figure 3 Class Diagram Visibility

Associations:

Associations represent static relationships between classes.

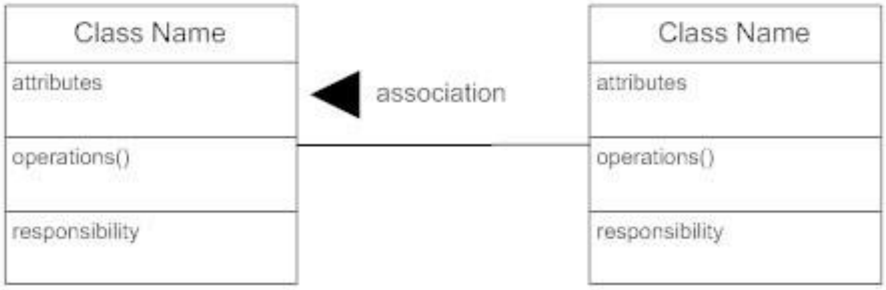


Figure 4 Class Diagram association

Active Classes:

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes.

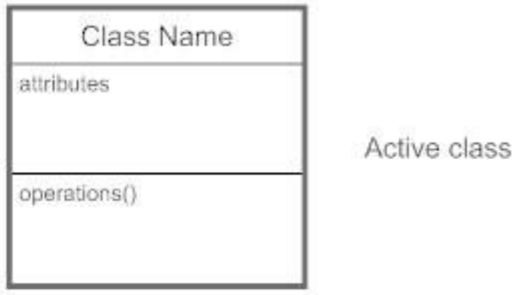


Figure 5 Class Diagram Active Class

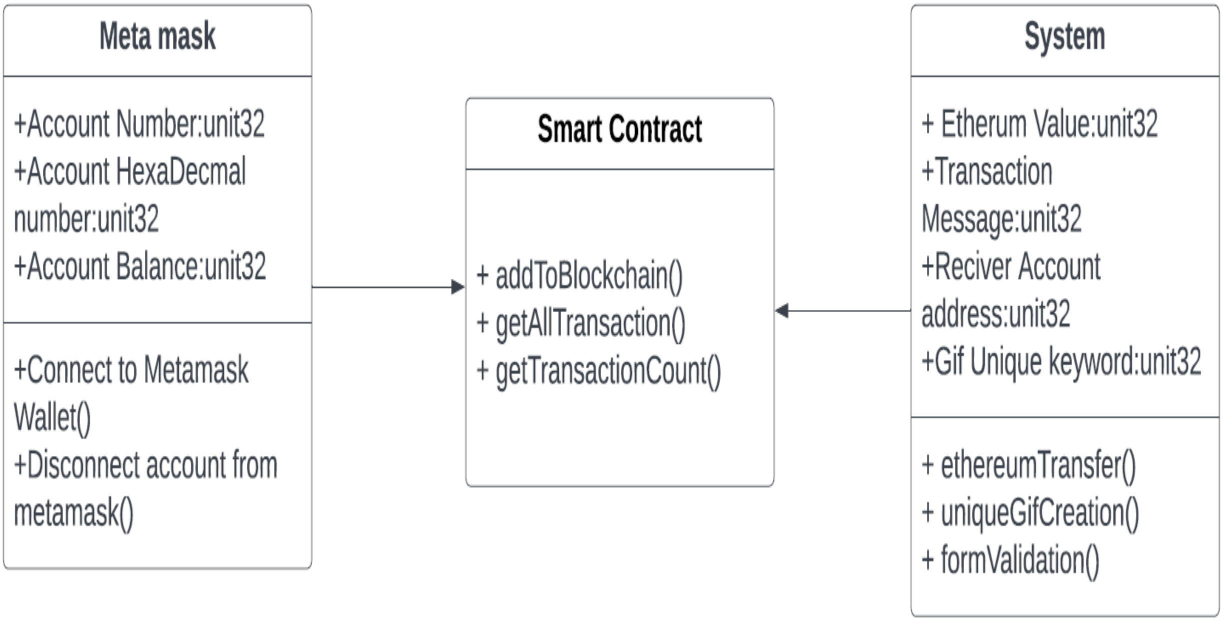


Figure 6 Class Diagram

3.3.2 Activity Diagram:

We use Activity Diagram to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We modal sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagrams focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. UML models basically three types of diagrams namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a Behavioral Diagram.

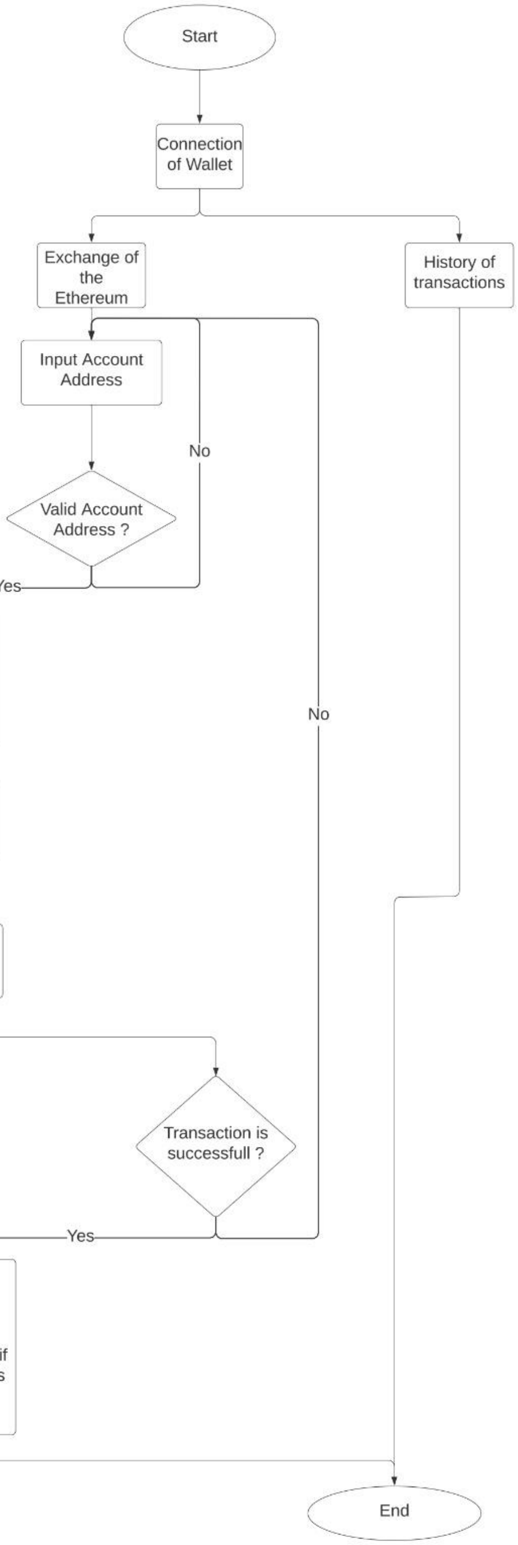


Figure 7 Activity Diagram

3.3.3 Use-case Diagram:

Use case diagrams are a common way to communicate the major functions of a software system. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Use cases are nothing but the system functionalities written in an organized manner. Nowanother thing which is relevant to the use cases are the actors. Actors can be defined assomething that interacts with the system.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Symbols used in Use Case diagram:

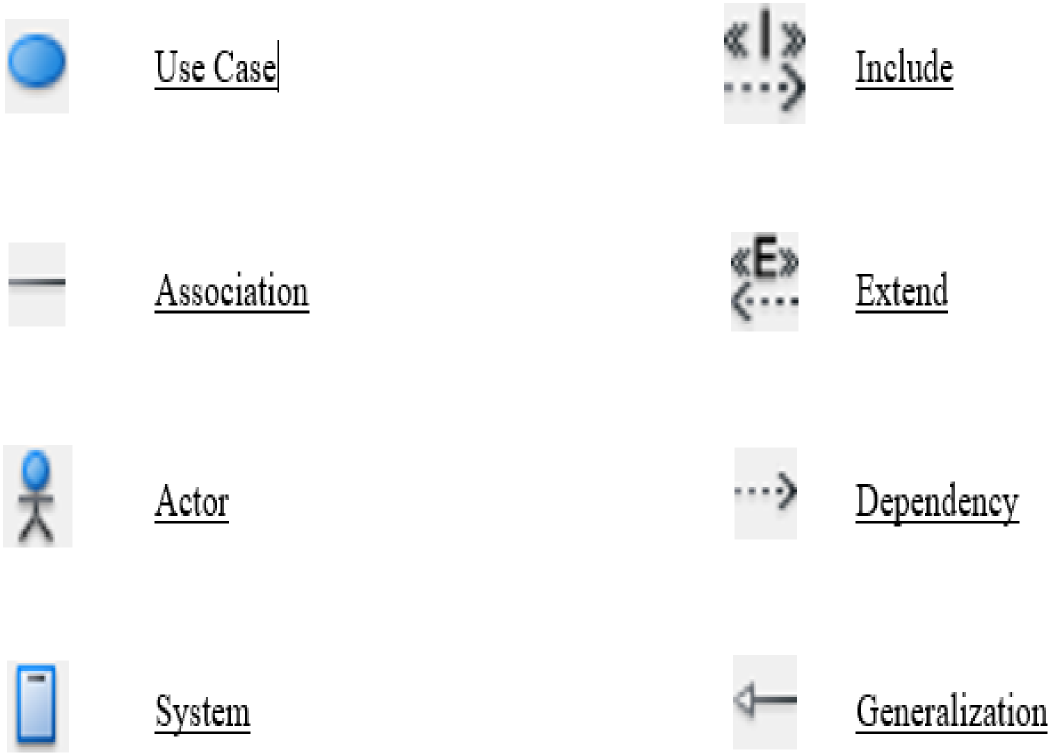


Figure 8 Use Case Diagram Symbol

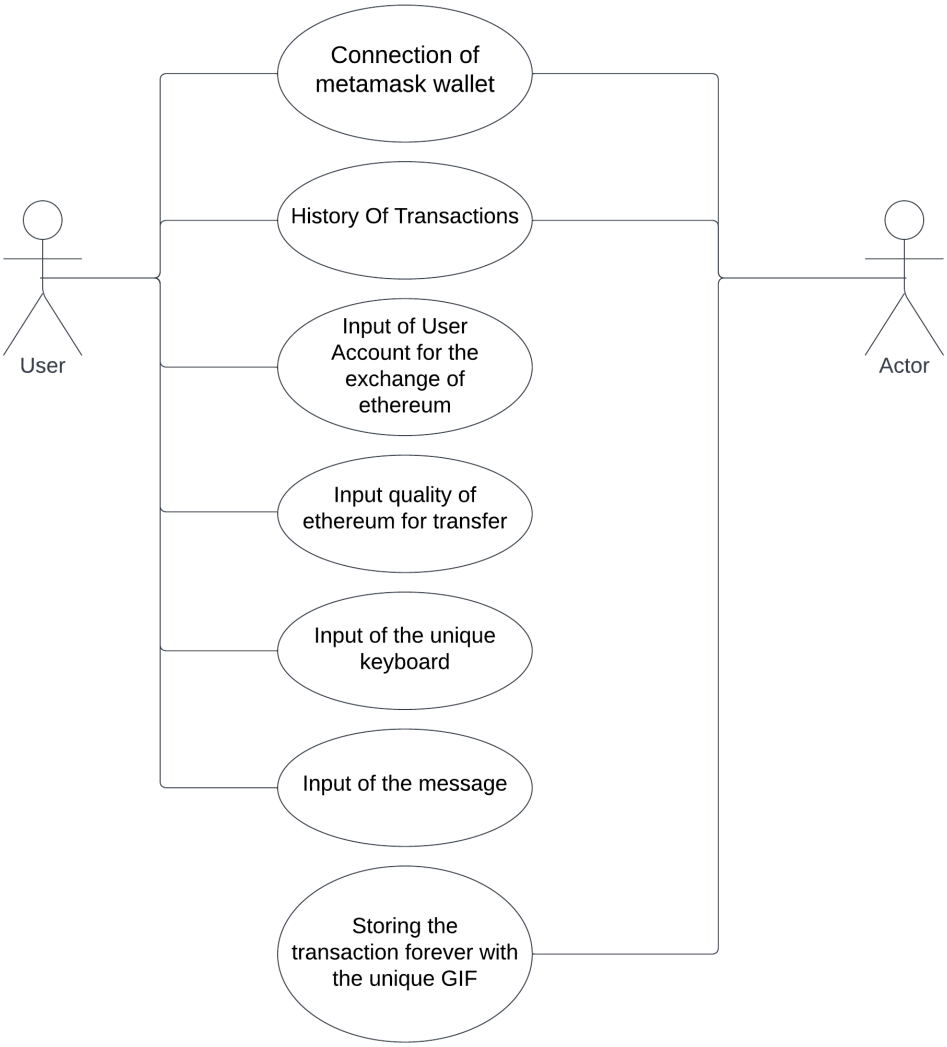

Connection of metamask wallet
History Of Transactions
Input of User Account for the exchange of ethereum
Input quality of ethereum for transfer
Input of the unique keyboard
Input of the message
Storing the transaction forever with the unique GIF
User
Actor

Figure 9 Use Case Diagram

3.3.4 Sequence Diagram:

Sequence diagram represents the behavioural aspects of a system. Sequence diagram shows theinteractions between the objects by means of passing messages from one object to another withrespect to time in a system.

Sequence diagram contains the objects of a system and their life-line bar and the messages passing between them. Objects appear at the top portion of sequence diagram. Object is shown in a rectangle box. Name of object precedes a colon ‘:’ and the class name, from which the object is instantiated. The whole string is underlined and appears in a rectangle box. A down-ward vertical line from object-box is shown as the life-line of the object. A rectangle bar on life-line indicates that it is active at that point of time. Messages are shown as an arrow from the life-line of sender object to the life-line of receiver object and labelled with the message name.

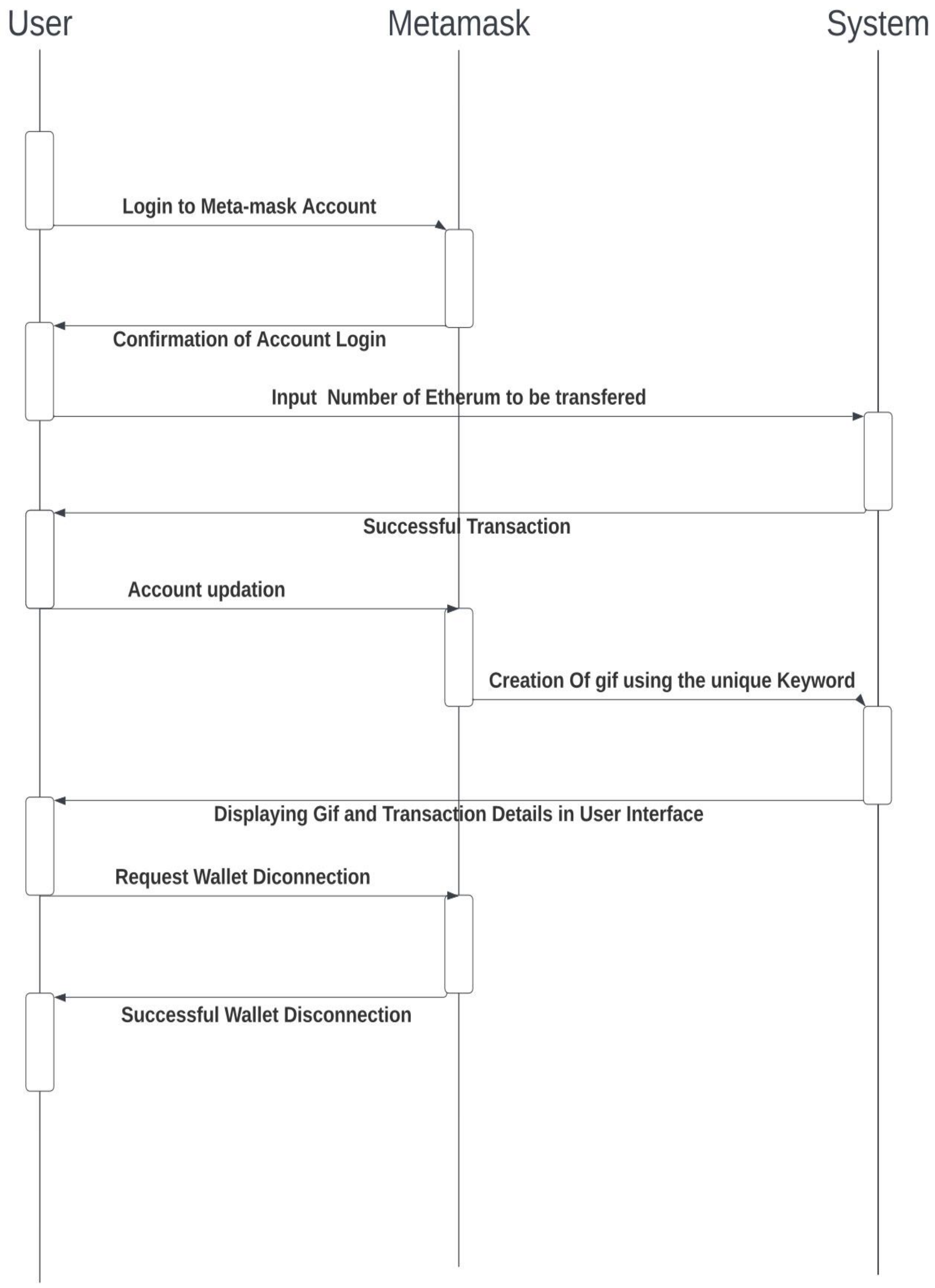

User
Metamask
System
Login to Meta-mask Account
Confirmation of Account Login
Input Number of Etherum to be transfered
Successful Transaction
Account updation
Creation Of gif using the unique Keyword
Displaying Gif and Transaction Details in User Interface
Request Wallet Diconnection
Successful Wallet Disconnection

Figure 10 Sequence Diagram

3.4 Methodology

3.4.1 Blockchain

Blockchain is a series of immutable blocks placed in chronological order of their mining. Blocks are packages of data whose value after mining can't be changed. All the blocks are chinned together with help of Cryptography Hash Function.

- Immutable:- No participant can change the data once it is recorded in Blockchain.
- Distributed Ledger Technology:- Distributed ledger technology(DLT) is a digital system for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time.
- Decentralization:- Decentralization refers to the transfer of control and decision-making from a centralized entity (individual, organization, or group thereof) to a distributed group.

Inefficiencies in the present-day systems are due to one major factors – presence of too many intermediaries. Consider a Cross Border Payment System (SWIFT System) for example.

There are can be many intermediaries who eat up a share of the price we pay for products. Let's look at this in more detail.

Suppose Bob who have a bank account in Bank of America and wants to send $100 to Alice in HNBC. When Bob is transferring the amount, they will provide the SWIFT code to Bank HNBC. SWIFT will then send a secure message communication to Bank HNBC including all necessary information. When Bank HNBC receives this message payment transfer will get started. Let's see how the transfer will work. SWIFT works when banks have a direct connection to each other meaning Bank of America will have their account in HNBC and Bank HNBC will have their account in Bank of America.

- Bank of America will deduct the $100 from Bob's account.
- Bank of America will credit the $100 to their account in HNBC.
- HNBC will deduct the $100 from Bank of America account in their bank.
- HNBC will credit the $100 (excluding fee) to Alice's account in their bank.

Now, there may be case that, Bank of America may not have direct connection in HNBC. i.e., BOA may not have account in HNBC. In such case BOA will use intermediator Banks to fulfil payment. In such case intermediator banks will also charge their fees. According to world bank the transfer charge is **around 7%** of the Actual Money Transfer Order.

After seeing what is Blockchain and why we need Blockchain, let’s look at the fundamentals of Blockchain. In that first is Blocks.

3.4.2 Blocks

A Blockchain is what it sounds like, it could be a chain of pieces containing a few information. The data can be anything like exchange information, code, or basic message. A Block consists of header, body, and hash of the past piece. To finalize a square a hub needs to illuminate a perplex, which is “Troublesome to unravel and simple to confirm” any hub can effortlessly verify the piece and can include the piece to the blockchain on coming to agreement.

The hash of the previous block makes sure that tempering any block will lead to recalculating the hash for all the next blocks which are difficult. Since all the blocks are distributed each peer in the network have a copy of the blockchain even though if someone solves the puzzle for all the next block it won't be possible to change the data in the blockchain because this copy can only be accepted only if 51% of the peers agree on the false blockchain.

3.4.3 Nodes

A blockchain comprises of various pieces of information. These information pieces are put away on nodes, which are comparable to little servers. On a blockchain, all hubs are connected to one another and constantly trade the foremost later data on the blockchain with one another. This ensures that all hubs are up to date.

Blockchain hubs are organize partners and their gadgets that are approved to manage the dispersed record and act as communication centre points for different arrange assignments. The primary work of a blockchain hub is to approve the legitimateness of each consequent batch of arrange exchanges, known as pieces.

There are two types of Nodes:-

1. Full Node: A full node is responsible to maintain all the transaction in blocks and add a valid block to the Blockchain.
2. Light Node:- A light node stores and provide data to accommodate daily activity and fast transaction.

3.4.4 Consensus Algorithm

Everything in the world requires a consensus for e.g. going for a trip we might ask a group of friends to vote for a place and we decide to move to a place that has the majority. Or one might appoint a person to decide on behalf of everyone and everyone agrees on the decision made by the person appointed that's consensus in real life.

The Blockchain arrange comprises of a arrangement of hubs that frame a dispersed architecture. These hubs ought to be adjusted and run synchronously to preserve security within the network. Thus the concept of Agreement is connected to preserve concordance within the Blockchain arrange.

Proof of work (used by Bitcoin and Ethereum) and Proof of Stake (Used by Solana and Polygon, Ethereum is planning to move to proof of stack after merge.)

In Proof of work, Nodes does extensive work to mine a block. i.e., Proof of Work requires lots of electricity. In case of PoW Nodes tries to maximize its CPU Power to increase their chance of validating and subsequently earn mining rewards. This leads to pooling and increase the chances of Centralization. Bitcoin consumes 1100 MW in total, that is 9636 GWh over an entire year, or 0.829 Mtoe.

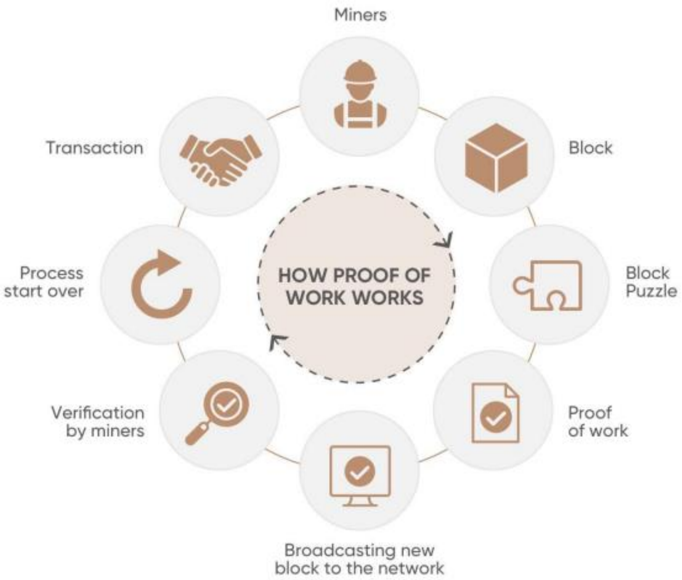


Figure 11 Proof of Work

After Proof of Work, let’s see Proof-of-Stake algorithms achieve consensus by requiring users to stake a number of their tokens to have a chance of being selected to validate blocks of transactions and get rewarded for doing so. PoS shares many similarities with PoW but also differs in fundamental ways. Every validator must own a stake in the network. Staking involves depositing some tokens into the systems, locking them in what you can think of like a virtual safe, and using it as collateral to vouch for the block.

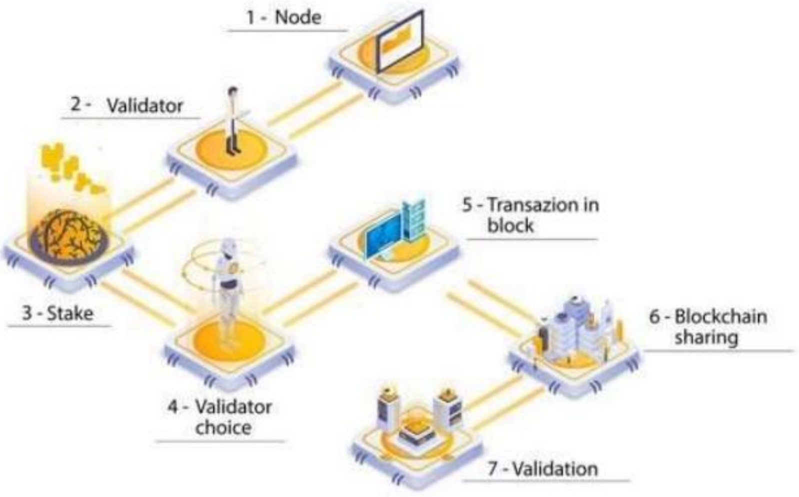


Figure 12 Proof of Stake

3.4.5 Wallets

A Cryptocurrency wallet could be a computerized holder that stores open and private keys for cryptocurrency exchanges. Since cryptocurrency may be a digital resource, it cannot be stored in a conventional wallet. A cryptocurrency wallet may be a gadget utilized to send and receive cryptocurrency. There are thousands of cryptocurrencies, and each has its unique supported wallet. No two cryptocurrency are same.

Wallet is mainly used for authentication and most Web 3.0 projects uses wallet to authorize users. Wallet will replace traditional method of remembering 100s of Id and Password. Some of famous wallets are: Meta mask, Trezor and Electrum.

3.4.6 Cryptocurrency Coins and Tokens

The term coin refers to any cryptocurrency that has its own separate, standalone blockchain (like Bitcoin, Ethereum, Polkadot, Cardano, etc.). The term token refers to any cryptocurrency that is built on top of an existing blockchain (e.g., ERC-20 tokens such as Maker (MKR), Basic Attention Token (BAT), USDT, Chain-link (LINK)).

Crypto currency is broadly classified into:

1. Store of Value: This kind of Currency are usually scare and are hedge against inflation. Bitcoin are said to be digital gold because it is scare in quantity i.e., 21 million.

2. Smart Contract: This kind of currency are used to pay gas fees and allow users to interact with smart contract. Ethereum is one of the widely used crypto currency for smart contracts. Polygon, Solana, Cardano and Tron are other few currencies in this category.

3. Digital Currency: This kind of currency are mostly designed to replace native currency and are planned to use for daily transaction. E.g., Ripple is alternative for cross border payment systems. Luna was another crypto currency project aimed to replace visa for merchant payment but failed lately due to Terra Luna – UST crash.

4. Exchange Tokens: This kind of tokens are mainly design to incentivize users who provide liquidity to the liquidity pools. Uniswap is one of the tokens used by Uniswap Liquidity Pools. Filecoin is one of the tokens used to incentivize users who allow the free space from their disk to blockchain to store and access data.

5. Utility Token: This kind of tokens are used for blockchain based service or products. Chainlink is one the project which provide off-chain data using oracles to the Blockchain smart contract.

6. Stable Coins: These coins are used in many DeFi projects the need for stable coin is that it provides stability unlike other volatile cryptocurrency the most common stable coins Tether which is collateralized stable coin but there are questions against do they have full backing for the Tether Supply. Recently Terra Luna's stable coin UST, which is algorithm stable coin failed to maintain its peg against USD and resulted in Death Spiral for the Luna Ecosystem (Market Cap from $ 68 Billion to $ 1.73 Billion). Most Stable coins aren't stable because of less backing of collateral or the coin is trying to maintain peg from another coin which result in death spiral. Currently USDC coin by circle and coinbase is having complete audits and DAI is said to have 1.5 times collateral to the native currency are the best stable coins.

7. NFTs/Collectibles: NFTs has been buzz word since last year. Basically, NFT Tokens can be used for royalty token or for something that is in fungible. NFTs and Collectibles are used in gaming and metaverse projects too. Some of famous NFTs are crypto kitties. NFTs are used for anything that isn't easily interchanged

3.5 BITCOIN

The Bitcoin whitepaper defines Bitcoin as a peer-to-peer electronic cash system. After the crash of 2008, an anonymous person or organization named Satoshi Nakamoto proposed a whitepaper for a currency that is not dependent on any centralized authority. The core idea behind Bitcoin was to use digital signatures as the solution to peer-to-peer value transfer without double-spending. The issue with the financial institution as they were inheriting the weakness of the trust-based model. Bitcoin proposed an electronic payment system based on cryptographic puzzles instead of trust. The paper gave a solution to the double-spending problem using peer-peer distributed systems. The system is secured long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

3.5.1 Transaction

By distributing transactional tasks among all computer devices, the cooperative network in which transactions take place is kept operational. When someone wishes to send bitcoins to another blockchain user, the network checks when the sender first received that amount (prior block) and certifies the amount being transferred to the recipient (future block). According to the network, doing so permanently transfers the money to the recipient and removes it from the sender's possession. To ensure that all transactions are verified and no room for fraud exists, the Bitcoin network made them public, allowing any user to access a record of transactions on the bitcoin blockchain. Each transaction is recorded in what the Bitcoin whitepaper refers to as a "timestamp," which shows where the amount existed previously and where it is going, thereby forming the chain of blocks.

3.5.2 Solving Double Spending

When digital assets and currencies first became available, the possibility of double spend the same asset arose. After all, P2P platforms existed more than a decade before bitcoin, allowing users to transfer files while also creating duplicates in their computers via a simple copy and paste process. What does bitcoin do to prevent this from happening? Every transaction in the blockchain is validated by all active nodes, who are running programmes to host and synchronise copies of the entire blockchain, in order to prevent double spending. The majority of modern computers can function as nodes, which aids the blockchain in validating transactions and blocks. The likelihood of double-spending is drastically decreased the more nodes the network has because every transaction is publicly reported, confirmed by nodes, and assessed by sender/receiver.

3.5.3 Proof of Work

A Proof of Work (PoW) system was introduced to demonstrate that the transactions are operating as intended. Each transaction in the PoW is assigned a random number that is connected to a small puzzle. The sender's system must complete the mathematical puzzle 19IT117 BLOCKCHAIN CORE CSPIT 15 KDPIT and send it to the receiver's system, which checks it into the chain, to complete the transaction. The deal is carried out after being shown to be accurate. Each transaction history is locked into blocks that pile up and get bigger as additional transactions are completed through the process of solving puzzles. Reversing transactions would require a tremendous amount of processing power; for example, 51% of all bitcoin's hash power would be needed to reverse one hour's worth of transactions. Those in charge of the operations are paid in bitcoin based on the quantity of valid transactions, which helps to ensure proper PoW. This is known as bitcoin mining.

3.5.4 Incentive

The idea behind the Bitcoin blockchain to keep it secure is to incentivize the miners to encourage them to stay honest. Miners are funded with an incentive in 2 ways:

- Transaction Fee: For all transaction users provide the transaction fee to include their block as quickly as possible. These fees are shared with miners.

- Mining Rewards: Miners are rewarded with some BTC for mining new blocks (initially it was 50 BTC) reduced every 4 years to a quarter (Currently 6.25 BTC). This mining reward is responsible for maintaining the circulation of new currency. Thus, the Bitcoin blockchain encourage miner to mine block and play by fair rules.

3.6 ETHEREUM

Ethereum Whitepaper was proposed by Vitalik Buterin, at the age of 19 to make Ethereum scripting language Turing complete. Vitalik believed that Bitcoin as a form of digital money is great but its scripting language is too weak.

3.6.1 Limitation of Bitcoin

- Lack of Turing completeness: Bitcoin scripting supports a large set of computations, but it doesn't support everything i.e., it lacks Loops. This is done to avoid infinite loops during transaction verification. One cannot code anything with 19IT117 BLOCKCHAIN CORE CSPIT 16 KDPIT Bitcoin script and it also increases unambiguous lines of Code. For eg. to implement ECSA 256 line of multiplication is written individually.

- Value Blindness: There is no way UTXOs script to provide fine-grained control over the amount that can be withdrawn.

- Lack of State: UTXO can either be spent or unspent there is no opportunity for multistage contracts or scripts which keep any state beyond.

- Blockchain-Blindness: UTXOs are blind to blockchain data such as the nonce and previous block hash. This limits several applications like gambling which requires a source of randomness from Blockchain data.

3.6.2 Ethereum Accounts

In Ethereum, the state is made up of object called accounts, with each account having 20 Bytes address and state transition being direct transfer of value and information between accounts. Fields of Ethereum Account:

- The nonce, a counter used to make sure each transaction can only be processed once
- The account's current ether balance
- The account's contract code, if present
- The account's storage (empty by default)

3.6.3 Message and Transaction

"Messages" in Ethereum are somewhat similar to "transactions" in Bitcoin, but with three important differences. First, an Ethereum message can be created either by an external entity or a contract, whereas a Bitcoin transaction can only be created externally. Second, there is an explicit option for Ethereum messages to contain data. Finally, the recipient of an Ethereum message, if it is a contract account, has the option to return a response; this means that Ethereum messages also encompass the concept of functions. Transaction are a signed data packed that stores a message to be send from an account. Transaction contains following details:

- message's recipient
- sender's signature
- ether to transfer
- data field
- STARTGAS: max steps an execution can take
- GASPRICE: fee the sender pays per step

3.6.4 State Transition

The Ethereum state transition function, APPLY(S, TX) -> S' can be defined as follows:

- Check if the transaction is well-formed (i.e., has the right number of values), the signature is valid, and the nonce matches the nonce in the sender's account. If not, return an error.

- Calculate the transaction fee as STARTGAS * GASPRICE, and determine the sending address from the signature. Subtract the fee from the sender's account balance and increment the senders nonce. If there is not enough balance to spend, return an error.

- Initialize GAS = STARTGAS, and take off a certain quantity of gas per byte to pay for the bytes in the transaction.

- Transfer the transaction value from the sender's account to the receiving account. If the receiving account does not yet exist, create it. If the receiving account is a contract, run the contract's code either to completion or until the execution runs out of gas.

- If the value transfer failed because the sender did not have enough money, or the code execution ran out of gas, revert all state changes except the payment of the fees, and add the fees to the miner's account.

Otherwise, refund the fees for all remaining gas to the sender, and send the fees paid for gas consumed to the miner.

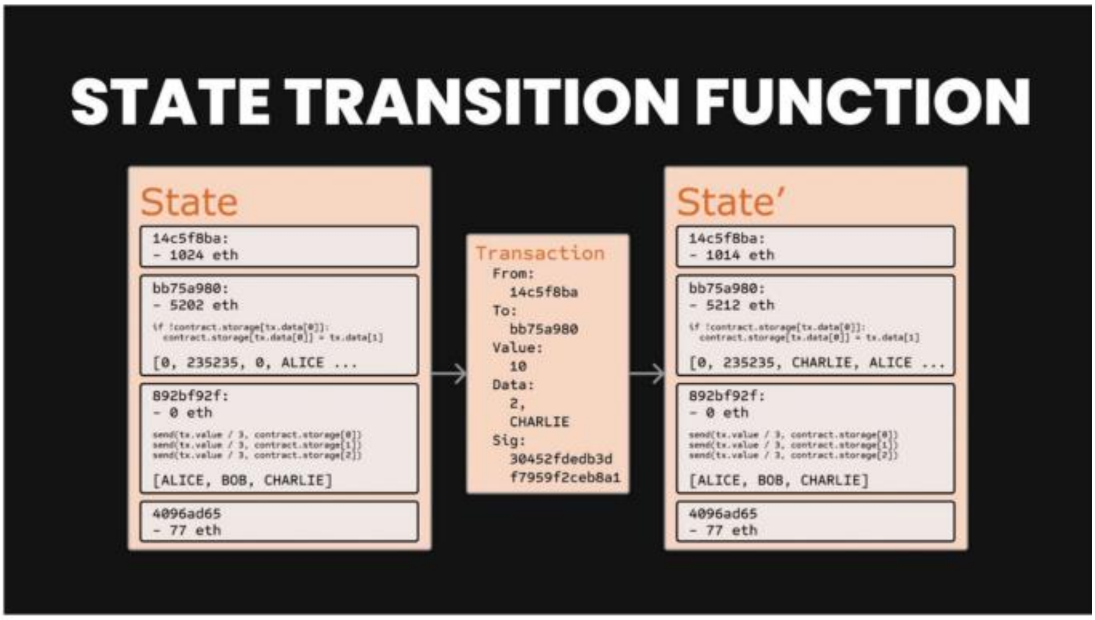

STATE TRANSITION FUNCTION
State
14c5f8ba:
- 1024 eth
bb75a980:
- 5202 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, 0, ALICE ...
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65
- 77 eth
Transaction
From:
14c5f8ba
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1
State'
14c5f8ba:
- 1014 eth
bb75a980:
- 5212 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, CHARLIE, ALICE ...
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65
- 77 eth

Figure 13 State Transition Function

3.6.4 Code Execution

The code in Ethereum contract is written in low-level, stack-based bytecode language, referred to as EVM code.

It's an infinite loop of:

- carrying out the operation at the current program counter
- incrementing program counter by one, This until occurs:
- end of the code
- error STOP/RETURN instruction

3.6.5 Mining Centralization

Ethereum address the mining centralization problem in Bitcoin mining and propose solution to it. Mining centralization can occur in following ways :

- Mining ecosystem has been dominated by ASICs (Application Specific Integrated Circuits), designed for mining and therefore are thousands of times more efficient in Bitcoin mining. This means that bitcoin mining is no longer a highly decentralized and require millions of dollars of capital to effectively participate.
- Bitcoin mining do not perform block validation locally instead they rely on centralized mining pool. This problem is arguably the worst as of now (in 2014) top 2 mining pools indirectly control 50% of mining and can mitigate 51% attack.

Thus, Ethereum tries to overcome the short comings of Bitcoin Blockchain and aims to build a blockchain that allows other token to work on top of it and allow code to execute using gas fees

3.7 Smart Contracts

A smart contract is a contract with additional blockchain features. It's a computer program or a transaction protocol to automatically execute, control, or document legally relevant actions/events according to some contract terms. Nick Szabo coined this term in 1990, referring to it as: "a set of promises, specified in digital form, including protocols within which the parties perform on these promises".

To understand the basic concept of what is a smart contract, we can make an analogy with a vending machine. By just inserting the coin and selecting what you need, it reduces the need for an intermediate to get your snack.

To get snacks from a vending machine: money + snack selection = snack dispensed

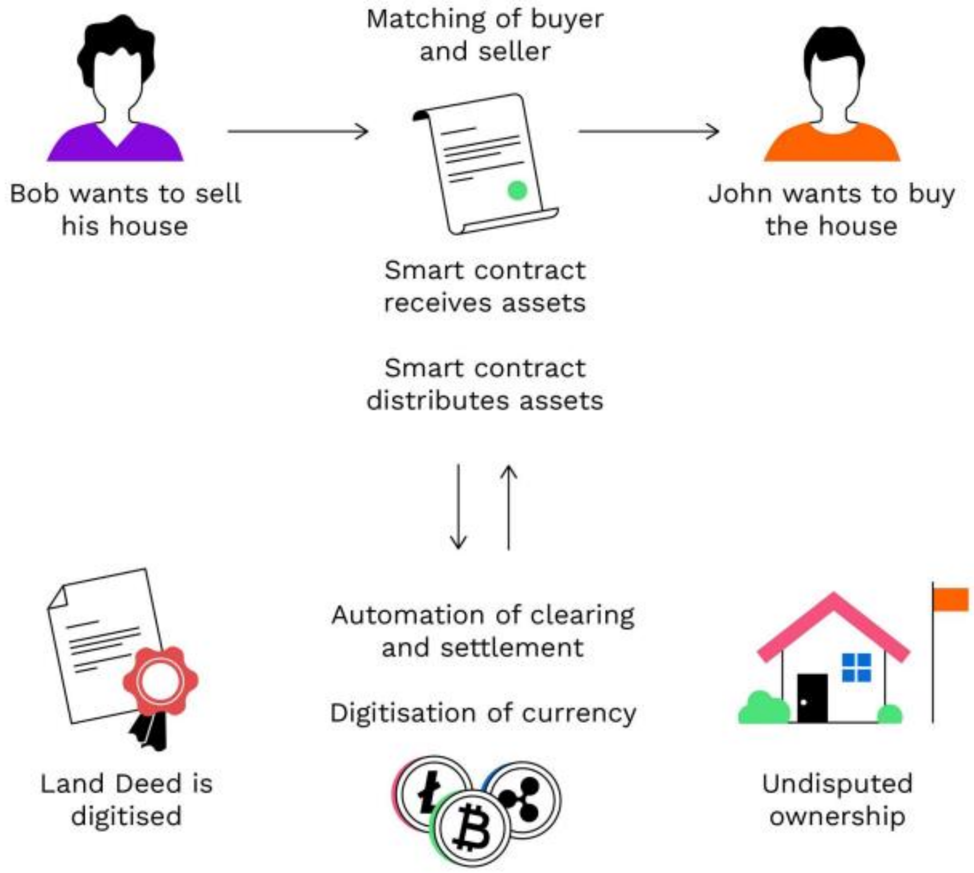

Matching of buyer and seller
Bob wants to sell his house
John wants to buy the house
Smart contract receives assets
Smart contract distributes assets
Automation of clearing and settlement
Digitisation of currency
Land Deed is digitised
Undisputed ownership

Figure 14 How do smart contract work

3.7.1 Smart Contract Languages

Smart contract can be written on multiple many blockchain one of the famous smart contract language is Solidity which is used by many blockchain including Ethereum, Binance Smart Chain , Polygon and Tron. Below is the list of Programming languages used by different blockchain. On Ethereum:

- Solidity
- Vyper
- Yul

On Solana:

- Rust

Other programming languages:

- JavaScript (Hyper ledger Fabric)
- Simplicity
- Scilla
- Ivy
- Bitcoin Script

3.7.2 Use Cases

Smart Contract and Insurance:

Insurance companies must have a formal agreement with the insured person that guarantees coverage according to the documented terms. Aside from that, they must manage claims resulting from various incidents and life events that trigger the agreement's activation. Smart contracts can handle the entire business process, from creating the formal agreement to issuing it to the insured and settling claims if the policy features need to be implemented.

Smart Contract in Financial Service:

Smart contracts are increasingly being used by banks and financial institutions to manage standard loans. Furthermore, syndicated loans, which involve multiple lenders providing loans to multiple borrowers on the same loan terms, stand to benefit greatly from the use of smart contracts. Using smart contracts, all steps, including syndication, diligence, underwriting, and servicing of syndicated loans, can be completed more quickly. In fact, with multiple entities involved in syndicated loans, establishing relationships, identities, and maintaining security becomes much easier with the on-chain / off-chain information that smart contracts can provide. In this scenario, smart contracts drive effectiveness and efficiency.

3.8 Timeline Chart

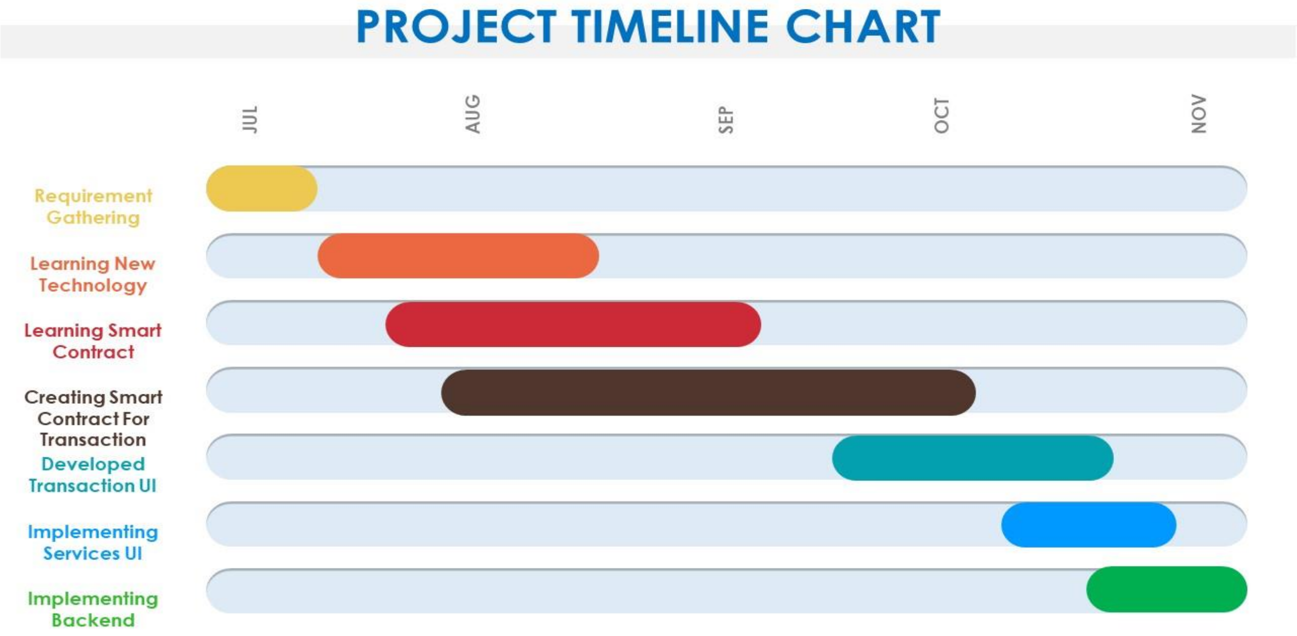
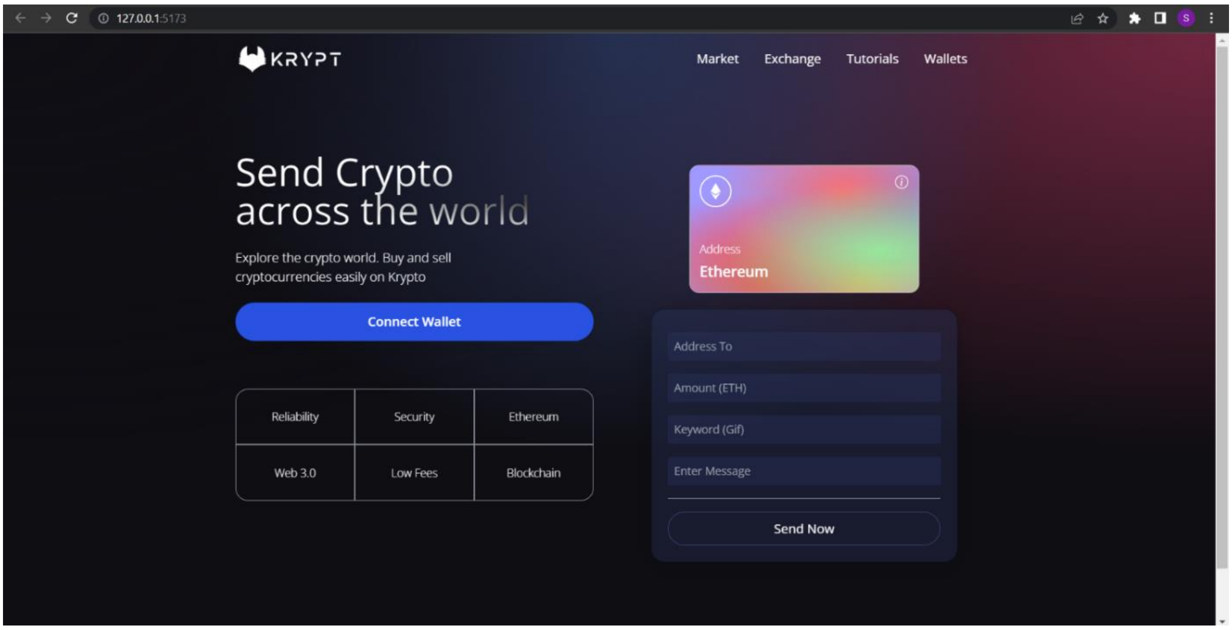


Figure 15 Project Timeline Chart

This is the month wise timeline chart of our work.

3.9 Template Design

As shown in figure 16, it is the home page of our application. There is a form on the home page. Users must fill out this form to make a transaction. This process may take 30 to 60 seconds to perform.



Send Crypto across the world

Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto

Figure 16 Transaction UI

Chapter 4: Implementation and Testing

4.1 Software and Tools :-

Hardware Requirements:-

Hardware that supports web app (E.g., Mobile, Computer)

Hardware must be there until it is hosted to cloud servers and ready to pitch. After that anyone can use it through any smartphones or pc that has web browser inside.

Software Requirements:-

- Front-End: React-js
  - React is a JavaScript library for building user interfaces.
  - React is used to build single-page applications.
  - React allows us to create reusable UI components.
- Back-End: Solidity
  - Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behavior of accounts within the Ethereum state.
  - Solidity is a Curly bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript. You can find more details about which languages Solidity has been inspired by in the language influences section.
  - Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.
  - With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.
  - When deploying contracts, you should use the latest released version of Solidity. Apart from exceptional cases, only the latest version receives security fixes. Furthermore, breaking changes as well as new features are introduced regularly. We currently use a 0.y.z version number to indicate this fast pace of change

- Others: Domain name, Hosting, sever and cloud storage

This web application needs to be computer under solidity language as it is the Blockchain technology supported as well the latest front-end language React-js that should be pre-installed in hardware in complete working position.

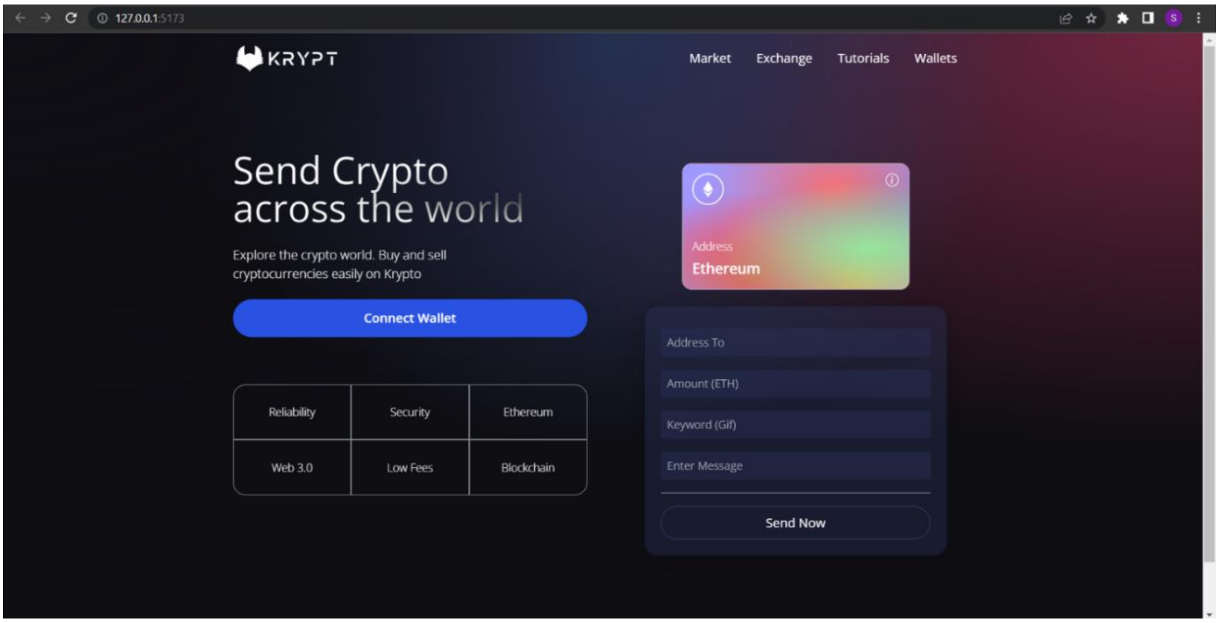
Software requirements for our clients:-

- Windows 7 or higher OS
- Google chrome or any other safe browser

Client just need not to worry about anything just trust on over new innovation and head towards the transaction through any smartphone or pc that has web browser inside.

4.2 User Interface and Snapshots

Figure 17, shows the home page of our application. The home page is the module that will appear first.

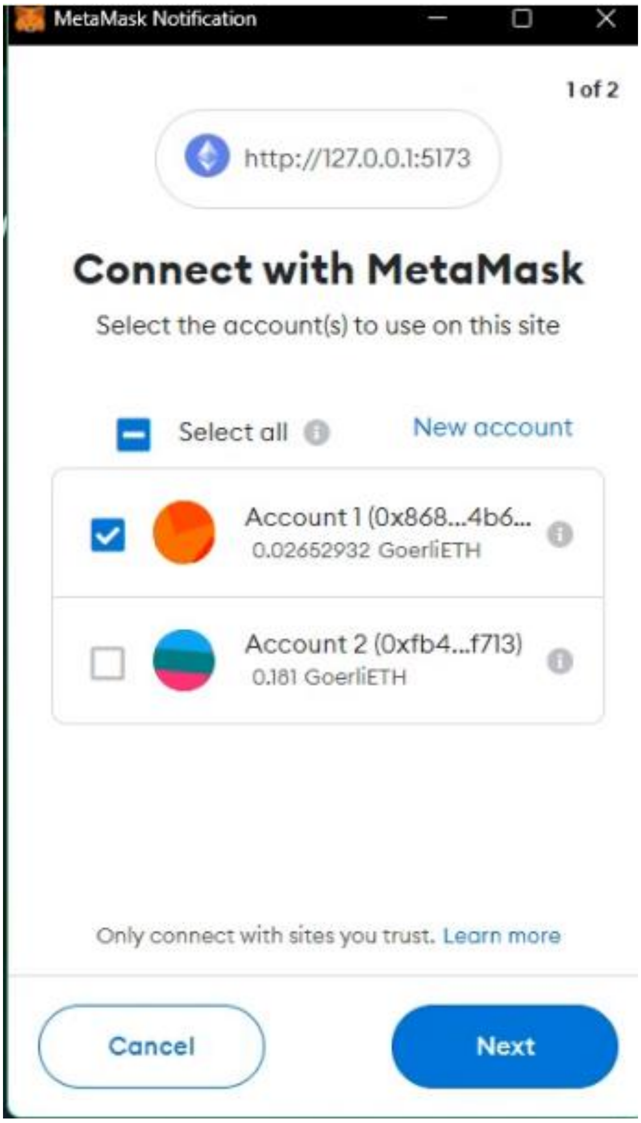


Send Crypto across the world

Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto

Figure 17 Home page of Application

As shown in figure 18, if a user wants to make a transaction, then the first step is to connect with their metamask account.



MetaMask Notification

1 of 2

http://127.0.0.1:5173

Connect with MetaMask

Select the account(s) to use on this site

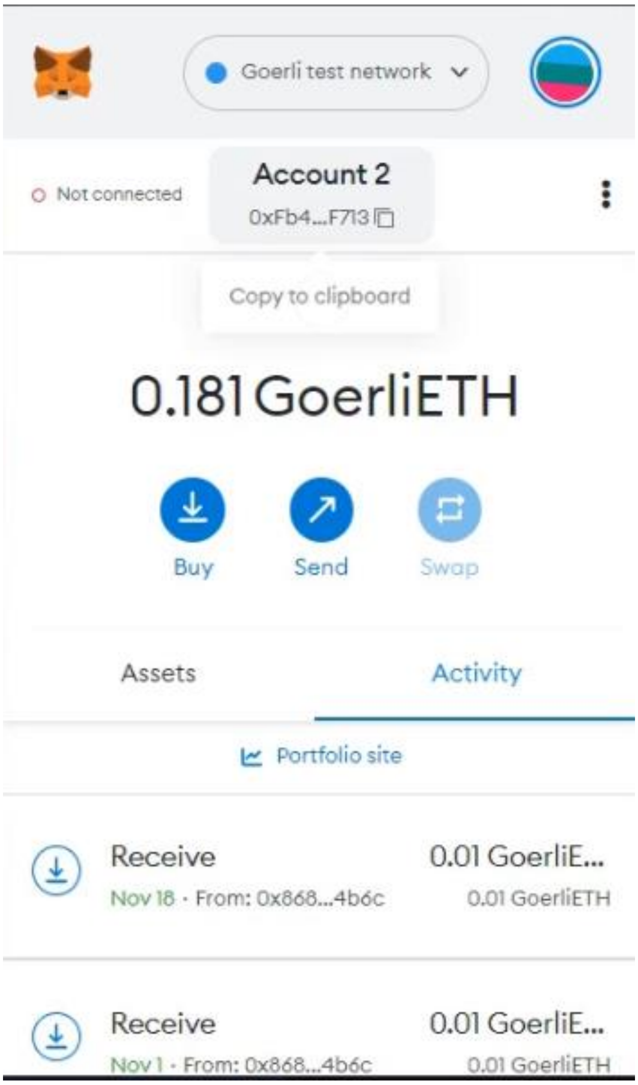
☑ Account 1 (0x868...4b6...
0.02652932 GoerliETH

□ Account 2 (0xfb4...f713)
0.181 GoerliETH

Only connect with sites you trust. Learn more

Figure 18 Connection with MetaMask

As shown in figure 19, this is a metamask account of a user. Users can see their account details and the history of transactions.



Goerli test network

○ Not connected

Account 2
0xFb4...F713

Copy to clipboard

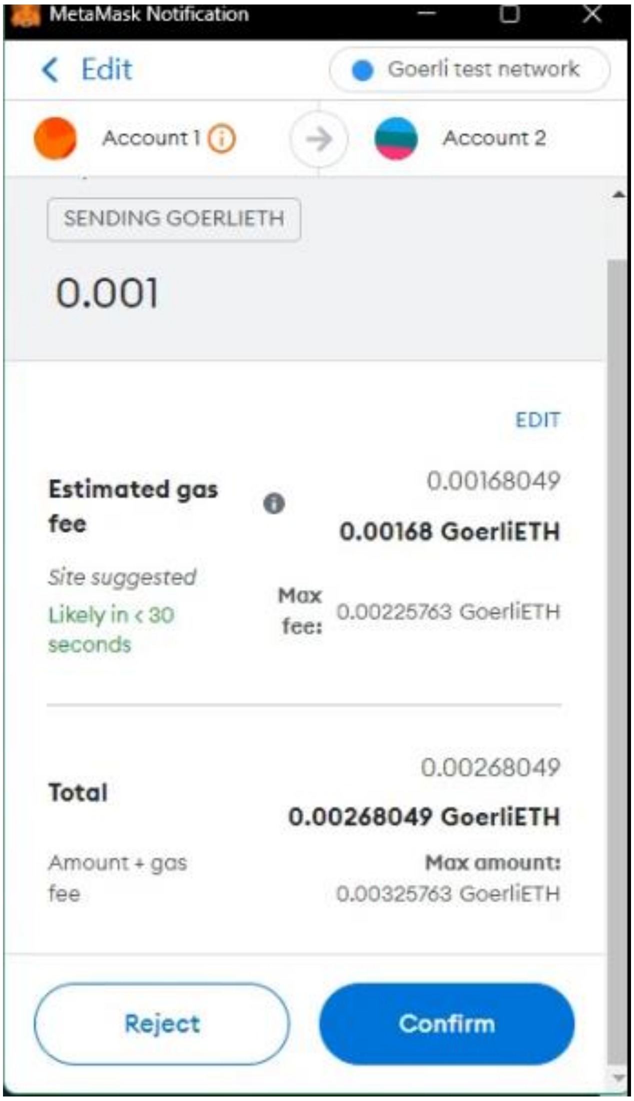
0.181 GoerliETH

Receive 0.01 GoerliE...
Nov 18 · From: 0x868...4b6c 0.01 GoerliETH

Receive 0.01 GoerliE...
Nov 1 · From: 0x868...4b6c 0.01 GoerliETH

Figure 19 Account of a User

As shown in figure 20, upon clicking on send now button, the system will ask a user to confirm the transaction. After clicking on confirm button the transaction process begins. It will take 30 seconds to 60 seconds to complete the transaction.



MetaMask Notification

Edit — Goerli test network

Account 1 → Account 2

SENDING GOERLIETH

0.001

EDIT

Estimated gas fee: 0.00168049 — 0.00168 GoerliETH

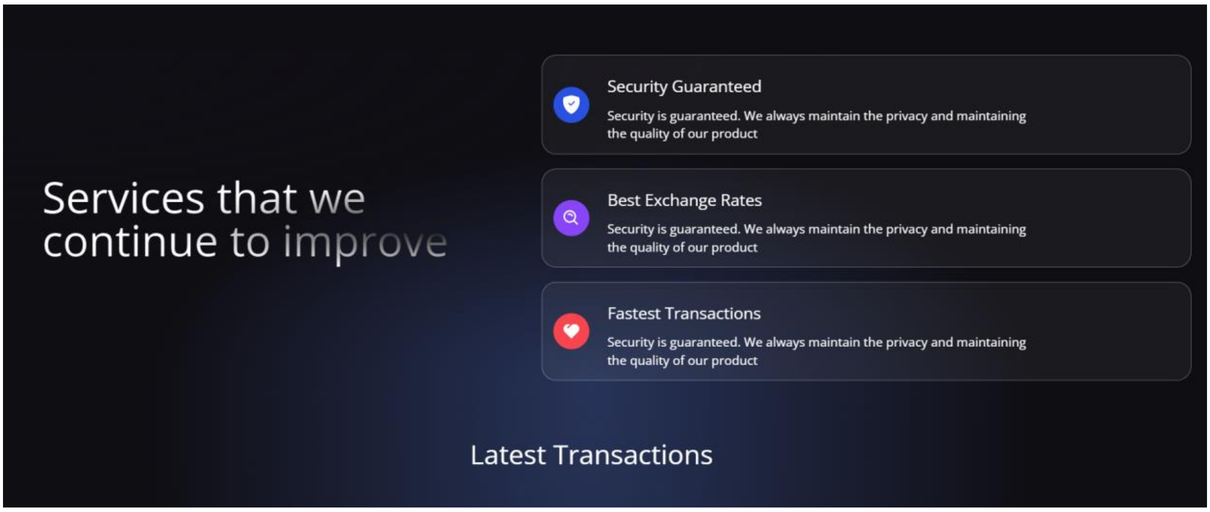
Site suggested — Likely in < 30 seconds — Max fee: 0.00225763 GoerliETH

Total: 0.00268049 — 0.00268049 GoerliETH

Amount + gas fee — Max amount: 0.00325763 GoerliETH

Figure 20 Confirmation Page

As shown in figure 21, this is the service page. On this page, we provide the services that are listed on the page for a smooth user experience.


Services that we continue to improve

Security Guaranteed
Security is guaranteed. We always maintain the privacy and maintaining the quality of our product

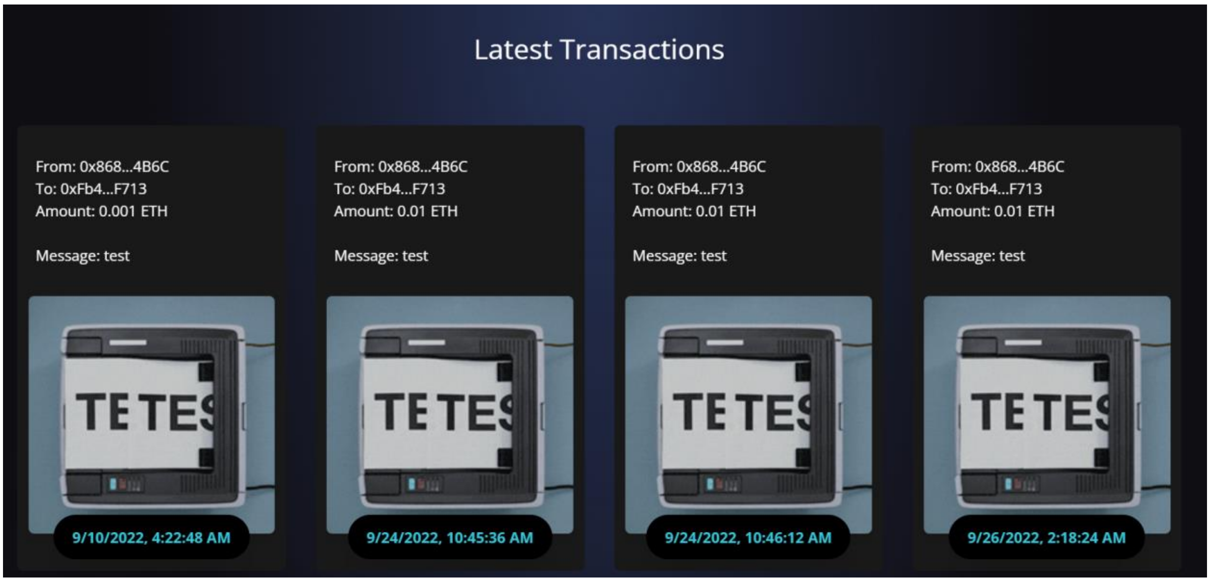
Best Exchange Rates
Security is guaranteed. We always maintain the privacy and maintaining the quality of our product

Fastest Transactions
Security is guaranteed. We always maintain the privacy and maintaining the quality of our product

Latest Transactions

Figure 21 Services of Platform

As shown in figure 22, On this page the user can see the list of all the transactions. Users can see all the transaction details like address, amount, message, GIF, date, and time.


Latest Transactions

From: 0x868...4B6C
To: 0xFb4...F713
Amount: 0.001 ETH

Message: test

9/10/2022, 4:22:48 AM

From: 0x868...4B6C
To: 0xFb4...F713
Amount: 0.01 ETH

Message: test

9/24/2022, 10:45:36 AM

From: 0x868...4B6C
To: 0xFb4...F713
Amount: 0.01 ETH

Message: test

9/24/2022, 10:46:12 AM

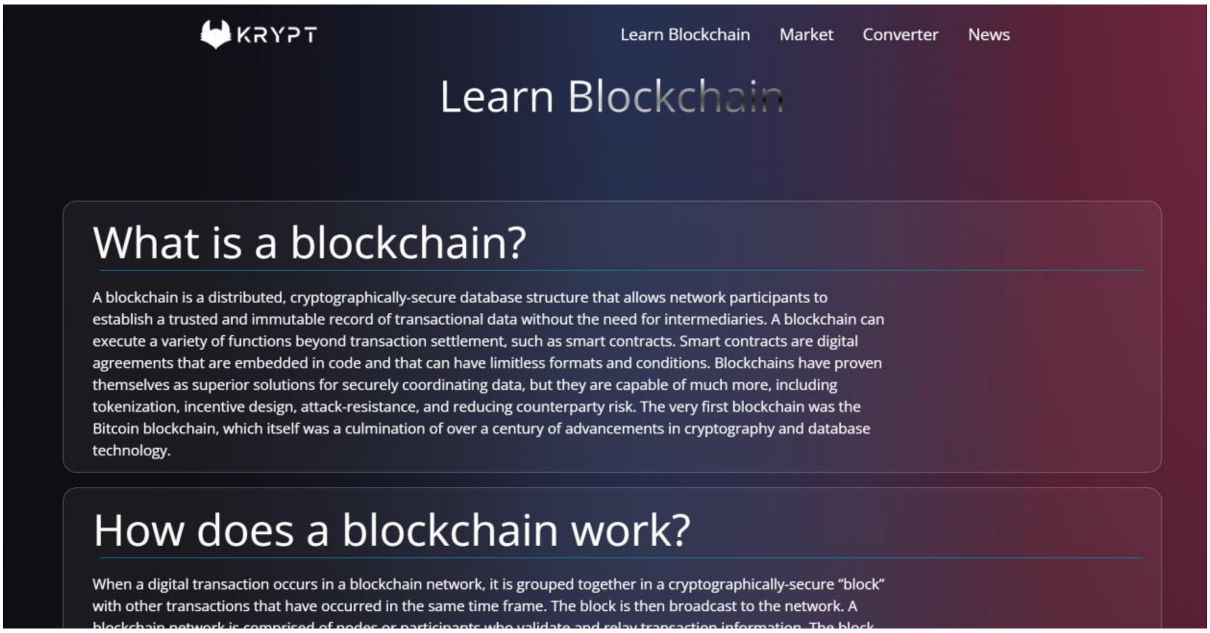
From: 0x868...4B6C
To: 0xFb4...F713
Amount: 0.01 ETH

Message: test

9/26/2022, 2:18:24 AM

Figure 22 Display Latest Transaction

As shown in figure 23, this is a frequently asked questions page. If users want to understand Blockchain and its fundamentals, they can visit this module. In this material, we provide the basic terms of Blockchain.



Learn Blockchain

What is a blockchain?

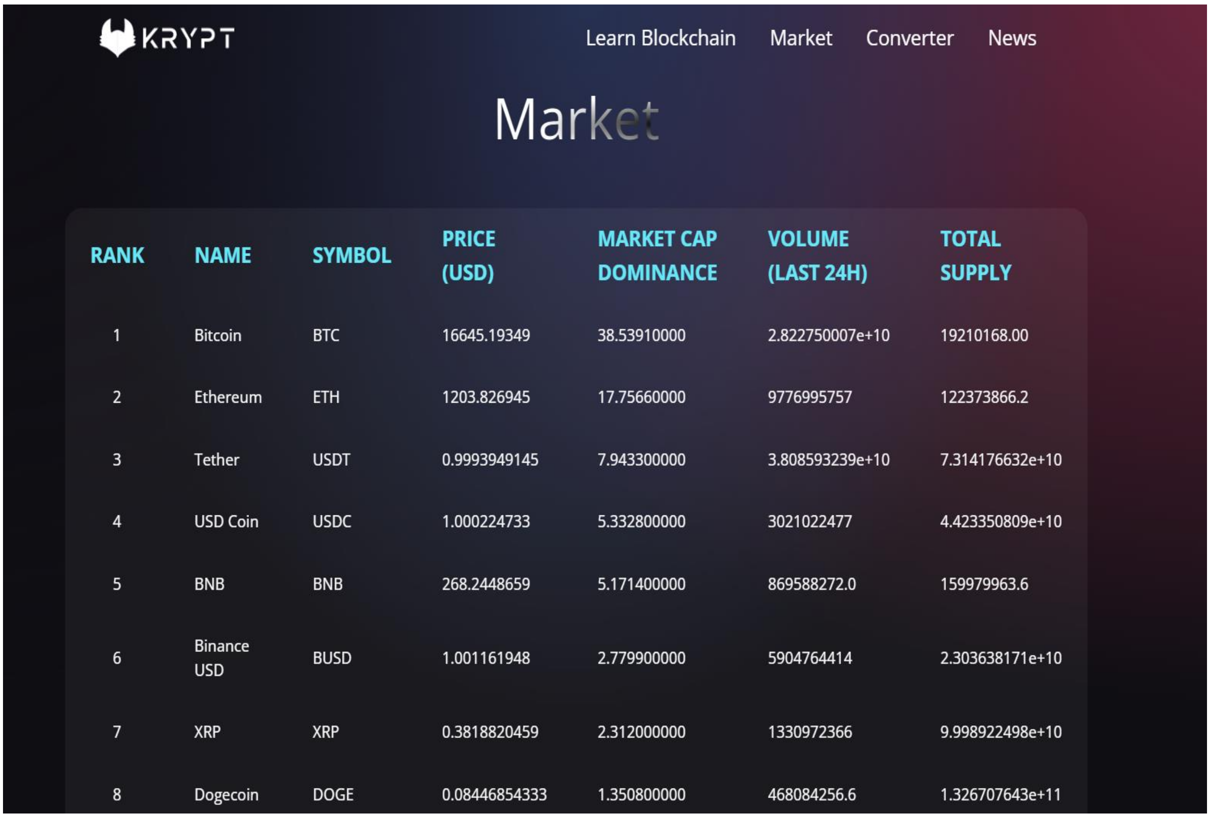
A blockchain is a distributed, cryptographically-secure database structure that allows network participants to establish a trusted and immutable record of transactional data without the need for intermediaries. A blockchain can execute a variety of functions beyond transaction settlement, such as smart contracts. Smart contracts are digital agreements that are embedded in code and that can have limitless formats and conditions. Blockchains have proven themselves as superior solutions for securely coordinating data, but they are capable of much more, including tokenization, incentive design, attack-resistance, and reducing counterparty risk. The very first blockchain was the Bitcoin blockchain, which itself was a culmination of over a century of advancements in cryptography and database technology.

How does a blockchain work?

When a digital transaction occurs in a blockchain network, it is grouped together in a cryptographically-secure "block" with other transactions that have occurred in the same time frame. The block is then broadcast to the network. A blockchain network is comprised of nodes or participants who validate and relay transaction information. The block

Figure 23 Learn Blockchain

As shown in figure 24, this is the current market page. The user can see the current rate of different cryptocurrencies on this page. Users can see the latest details of the market.

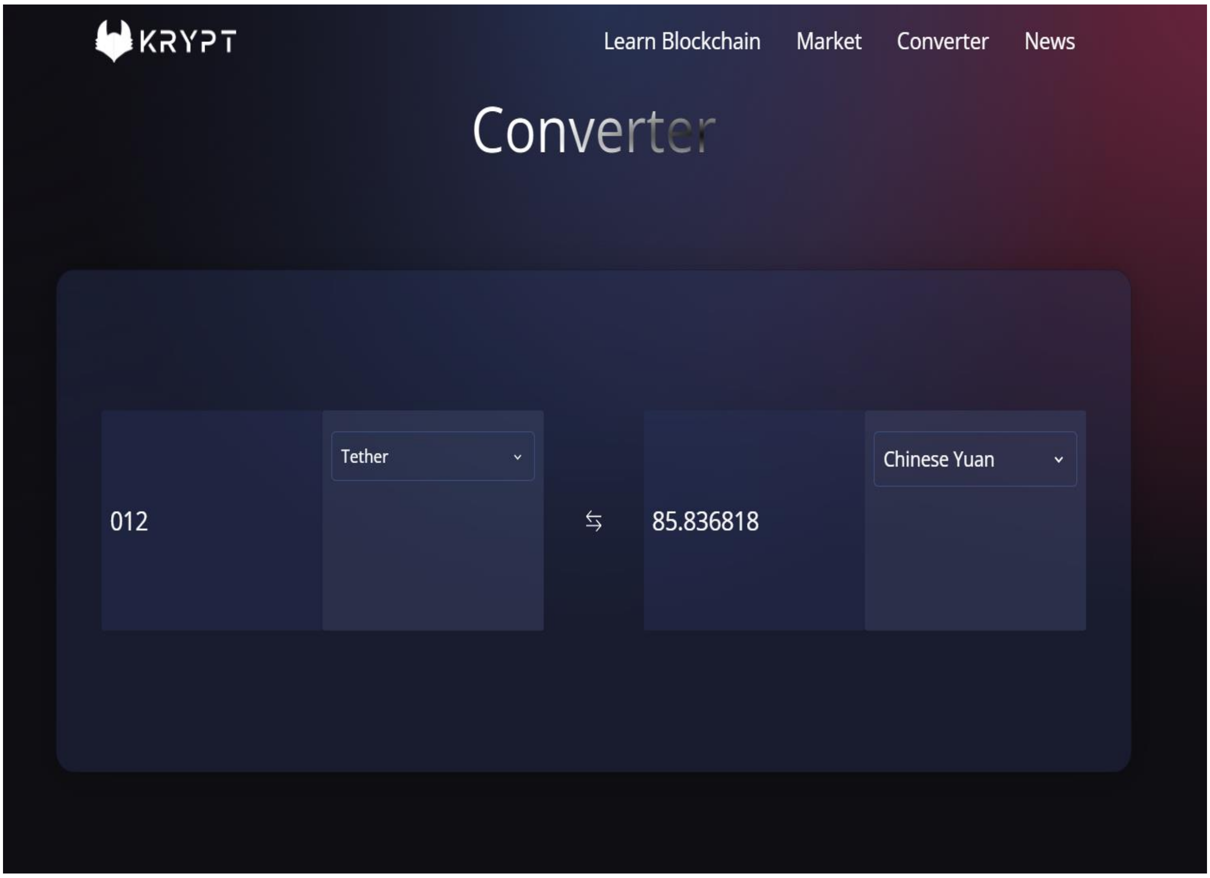


Market

RANK	NAME	SYMBOL	PRICE (USD)	MARKET CAP DOMINANCE	VOLUME (LAST 24H)	TOTAL SUPPLY
1	Bitcoin	BTC	16645.19349	38.53910000	2.822750007e+10	19210168.00
2	Ethereum	ETH	1203.826945	17.75660000	9776995757	122373866.2
3	Tether	USDT	0.9993949145	7.943300000	3.808593239e+10	7.314176632e+10
4	USD Coin	USDC	1.000224733	5.332800000	3021022477	4.423350809e+10
5	BNB	BNB	268.2448659	5.171400000	869588272.0	159979963.6
6	Binance USD	BUSD	1.001161948	2.779900000	5904764414	2.303638171e+10
7	XRP	XRP	0.3818820459	2.312000000	1330972366	9.998922498e+10
8	Dogecoin	DOGE	0.08446854333	1.350800000	468084256.6	1.326707643e+11

Figure 24 Current Market

As shown in figure 25, this is the converter page. Users can convert different cryptocurrencies to different national currencies.

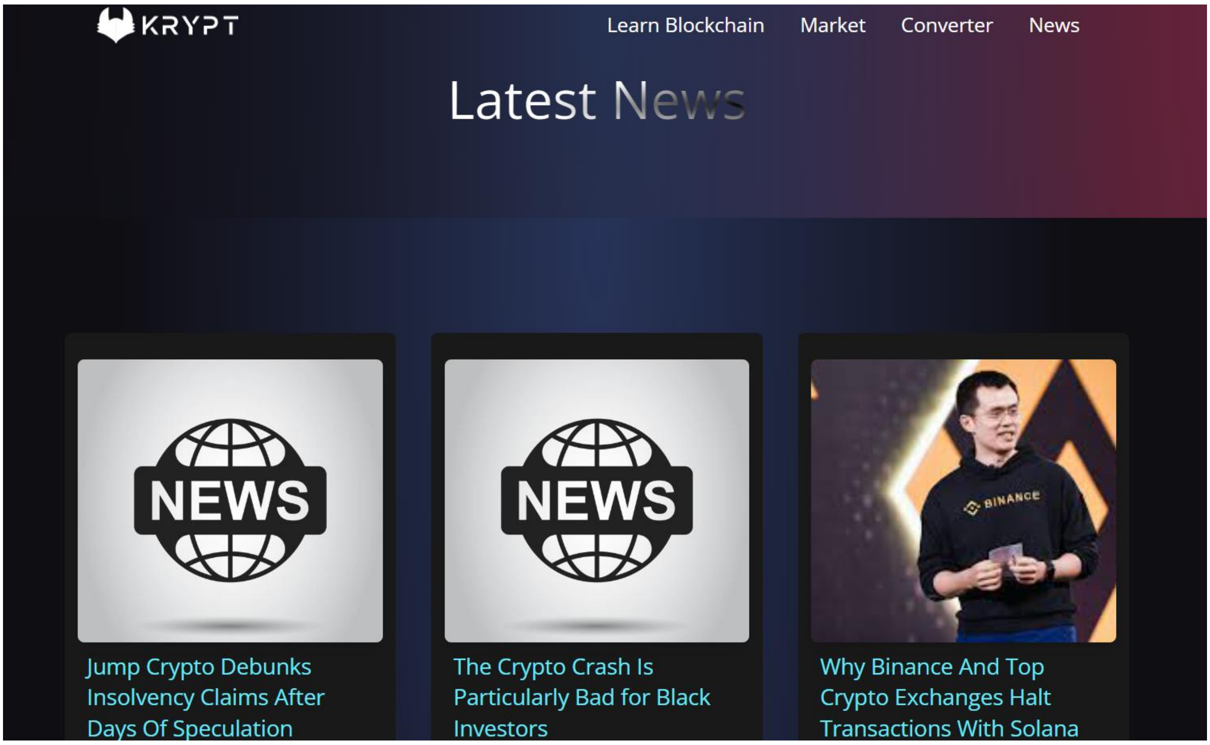


012

85.836818

Figure 25 Crypto Currency Convertor

As shown in figure 26, this is a news page. Users can see the latest news related to Blockchain and cryptocurrency. This module is helpful for the user to stay updated.



Latest News

Jump Crypto Debunks Insolvency Claims After Days Of Speculation

The Crypto Crash Is Particularly Bad for Black Investors

Why Binance And Top Crypto Exchanges Halt Transactions With Solana

Figure 26 Latest new for Crypto

Chapter 5: Conclusion and Future work

5.1 Conclusion

The proposed Blockchain based transaction application manages all the Ethereum records. With the help of Blockchain, we can done transaction with more security and transparency. If we do not have details of account then, we cannot get details back.

Here, we have applied maximum qualities of blockchain technology and implement a transaction of Ethereum which will solve the transactional problem of Ethereum and make the work easier and faster.

Blockchain technology has the potential to be implemented in a far more secure and accessible transaction system. In future, we believe that Blockchain based transaction system can replace any transaction system.

5.2 Future work

Implementation of different cryptocurrencies transaction. Exchange rate of different cryptocurrencies will be provided for better results.

Decreasing the usage of gas for each transaction. Creating more smart contracts for better security. News of cryptocurrencies will be provided so the users will stay updated

Chapter 6: References

1. “Remix - Ethereum IDE.” Remix - Ethereum IDE, remix.ethereum.org. Accessed 18 Nov. 2022.

2. “Introduction | MetaMask Docs.” Introduction | MetaMask Docs, 14 Jan. 2022, docs.metamask.io/guide.

3. “Install Tailwind CSS With Vite - Tailwind CSS.” Install Tailwind CSS With Vite - Tailwind CSS, tailwindcss.com/docs/guides/vite. Accessed 18 Nov. 2022.

4. Barulli, M., Weigand, F., and Reboh, P. (2017). Bernstein - Product Deck - Blockchain Solutions for Securing Intellectual Property Assets and Innova-tion Processes, 1–14.

5. Onuklu, A. (2019), "Research on Blockchain: A Descriptive Survey of the Literature", Choi, J. and Ozkan, B. (Ed.) Disruptive Innovation in Business and Finance in the Digital World (International Finance Review, Vol. 20), Emerald Publishing Limited, pp. 131-148. DOI/10.1108/S1569-3767201

6. “8 Best Crypto Wallets of November 2022.” Money, money.com/best-crypto-wallets. Accessed 18 Nov. 2022.https://www.techtarget.com/searchcio/feature/Top-10-benefits-of-Blockchain-technology-for-business

7. Blockchain Nodes: How They Work (All Types Explained) - Nodes.com, n.d.

8. Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2017). “An overview of Blockchain technology: architecture, consensus, and future trends,” in 2017 IEEE International Con-gress on Big Data (BigData Congress) (Honolulu, HI), 557–564. doi: 10.1109/BigDataCongress.2017.85

9. Catalini, C., and Gans, J. S. (2016). Some Simple Economics of the Blockchain. Rotman School of Management Working Paper No. 2874598, MIT Sloan Research Paper No. 5191-16. Social Science Research Network (SSRN). doi: 10.2139/ssrn.2874598

10. L. Weese, “Bitcoin mining and energy consumption,” 8 December 2017. [Online]. Available:https://blog.bitcoin.org.hk/bitcoin-mining-and-energy-consumption4526d4b56186.

11. Codex (2018). Codex Protocol - A Cryptocurrency and Decentralized Registry for Unique Assets, Starting With Art & Collectibles.

12. “Ethereum - ETH Price, Live Chart, and News | Blockchain.com.” Ethereum - ETH Price,Live Chart and News|Blockchain.com www.blockchain.com/explorer/assets/[id].Accessed,18,-Nov.2022. https://www.bbvaresearch.com/wp-content/uploads/2016/12/WP_16-20.pdf

13. Pratt, M. K. (2021, June 2). Top 10 benefits of blockchain technology for business. SearchCIO.Retrieved,November,-18,2022,from https://www.techtarget.com/searchcio/feature/Top-10-benefits-of-blockchain-technology-for-business

14. V. Buterin, “Ethereum: A Next-Generation Smart Contract and Decentralized Application,Platform.,”.December,2014.[Online],.Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepape r_-_Buterin_2014.pdf. [Accessed 06 July 2022].

15. Pilkington, Marc, Blockchain Technology: Principles and Applications (September 18, 2015). Research Handbook on Digital Transformations, edited by F. Xavier Olleros and Majlinda Zhegu. Edward Elgar, 2016, Available at SSRN: https://ssrn.com/abstract=2662660.

16. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed 06 July 2022].