Krypt Web3.0

Blockchain Technology

By

8853 – Jayesh Badwal

8880 – Lidya Simon

8888 – Prachi Mohare

BE Comps A

Under the guidance of –

Prof. Monali Shetty

DEPARTMENT OF COMPUTER ENGINEERING

Fr. Conceicao Rodrigues College of Engineering, Bandra (W),

Mumbai – 400050

University of Mumbai

2022-23

Abstract

We implement a web3.0 application from scratch that allows users to make ethereum transactions through the blockchain network. Krypto is designed with Reactjs, a Javascript library. This framework is connected to blockchain technology to perform decentralized storage of transactions. Krypt Web3.0 will be then paired with metamask,a cryptocurrency wallet that enables users to store Ether and other ERC-20 tokens. We use goerli ethereum test network in order to perform dummy transactions.

Authenticated users will connect to the metamask ethereum wallet that is setup to work with the private blockchain. This wallet can be used to interact with decentralized applications, hence we pair it with our application. Then we write smart contracts on the ethereum network to process all transactions in a contract hence, middle men are not required for executing the transactions. Each transaction will then be paired with a gif and will be permanently stored in the blockchain network and transaction cards are generated with detailed view to the current transaction data as well as previous transaction records as a user interface. These applications include the transaction of Ethereum from one wallet to another or receive Ethereum from others. It also kept all the data of transaction details like how many Ethereum are sent to the receiver and timing of the transactions. This is one of the important features of our system.

Introduction

Each transaction in the blockchain is signed by the rightful. To order these transactions and prevent the double-spending problem, blockchain uses proof-of-work. The proof-of-work is a procedure that establishes a cost for grouping transactions in a certain order and adding them to the blockchain. These corporations of transactions are referred to as blocks. Each block points to a previous block in the chain, thus the name blockchain.

This is why cryptocurrencies, like Bitcoin, are constructed on blockchain technology. Blockchain offers cryptocurrencies the platform and protection they want to work. These blocks, or statistics are saved on on-line servers, which are publicly accessible, but only via means of you or anyone else who the block has been shared with. In our project we use web 3.0 which is a third generation of internet services which provide web sites and packages with the technology to run. Web 3.0 is ready to be powered via means of AI and peer-to-peer packages like blockchain. The key difference between Web 2.0 and Web 3.0 is that Web 3.0 is more focused on using innovative technologies like machine learning and AI to create greater personalized content for each user. It is also expected that Web 3.0 will be more stable than its predecessors because of the system it is constructed upon. Blockchain serves as the foundation of Web 3.0. Web 3.0 could not be feasible without blockchain. The enhanced protection and privacy presented with blockchain is something that the developers of Web 3.0 are using to appeal to internet users. Since blockchain is a decentralized system, there is no unmarried factor of control that could be easily hacked. This means the web sites and the internet, in general, could be much greater stable against attacks. Users could now no longer fear their statistics being deleted or compromised. Together, Web 3.0 and blockchain will permit for higher crypto currency trading and mining with security as its key feature.

Problem Statement

To build a full-fledged web3.0 application that allows users to make ethereum transactions through the blockchain. Each transaction is then paired with a gif and are permanently stored in the blockchain network.

Feature and flow

BLOCKCHAIN WALLET : A blockchain wallet is a cryptocurrency wallet that allows users to manage different kinds of cryptocurrencies—for example, Bitcoin or Ethereum. A blockchain wallet helps someone exchange funds easily. Transactions are secure, as they are cryptographically signed. The wallet is accessible from web devices, including mobile ones, and the privacy and identity of the user are maintained. So a blockchain wallet provides all the features that are necessary for safe and secure transfers and exchanges of funds between different parties.

1. Easy to use. It's just like any other software or a wallet that you use for your day-to-day transactions.
2. Highly secure. It is just a matter of securing your private key.
3. Allows instant transactions across geographies. And these are barrier-free, without intermediaries.
4. Low transaction fees. The cost of transferring funds is much lower than with traditional banks.
5. Allows transactions across Ethereum. This helps you do easy currency conversions.
6. Every node on the network has a copy of the digital ledger. To add a transaction every node needs to check its validity. If the majority thinks it's valid, then it's added to the ledger. This promotes transparency and makes it corruption-proof.

HARDHAT : Hardhat is a development environment to compile, deploy, test, and debug your Ethereum software. It helps developers manage and automate the recurring tasks that are inherent to the process of building smart contracts and Apps, as well as easily introducing more functionality around this workflow. This means compiling, running and testing smart contracts at the very core. A lot of Hardhat's functionality comes from plugins, and, as a developer, you're free to choose which ones you want to use. Hardhat is not opinionated in terms of what tools you end up using, but it does come with some built-in defaults. All of which can be overridden. It runs as either an in-process or stand-alone daemon, servicing JSON-RPC and Web Socket requests. By default, it mines a block with each transaction that it receives, in order and with no delay. It's backed by the

@ethereumjs/vm EVM implementation, the same one used by ganache, Remix and Ethereum Studio.

TAILWIND : Tailwind CSS is as told in its documentation is a utility-first CSS framework. With this it means that it doesn't have those predesigned elements and components. All you get with tailwind CSS is a bunch of classes which you can use in combination to create a beautiful UI. Tailwind CSS doesn't have these inbuilt pre-styled components. It will give you the classes and you can style it yourselves. For example the container in the tailwind just gives you a width with no padding and no margin. Tailwind also offers the best UI components but they are paid but still here are some of the best libraries or websites that offer pre built tailwind blocks. And the best part about them is first they all are responsive and secondly you don't have to add or install anything.

METAMASK : Metamask is a popular crypto currency wallet, surpassing 10 million monthly active users. It is a crypto wallet that you can use while browsing the web to interact with decentralized applications. It can store multiple private keys and can work on multiple networks, such as the Ethereum network and the Binance Smart Chain network. If you've been hearing news about Axie Infinity and other blockchain games, at one point you've probably heard of MetaMask too. It allows you to do transactions between decentralized applications like UniSwap, Pancake Swap, ShibaSwap, and so on, with just a few clicks. You can also use MetaMask to register on marketplaces. Imagine going to the mall to buy groceries, would you have to memorize your banking credentials to have access to your funds inside it? No, you have your credit/debit card for that. It's automatic, you will just need to swipe your card, type your PIN (if they require it), and voila! MetaMask works like that, making your life much easier, but for crypto.

VITE: - "Vite (French word for "fast", pronounced /vit/) is a build tool that aims to provide a faster and leaner development experience for modern web projects. It consists of two major parts: A dev server that provides rich feature enhancements over native ES modules, for example extremely fast Hot Module

Replacement (HMR). A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for production. Vite is opinionated and comes with sensible defaults out of the box, but is also highly extensible via its Plug-in API and JavaScript API with full typing support."

a. Project flow:

- Connect reactjs app to the blockchain.
- pair it to an ethereum wallet using metamask.
- Write smart contracts on the ethereum network using solidity programming language.
- Implementing web3.0 into react application makes it send transactions through blockchain.
- store those transactions with added details as a card in our application. Details include:
    1. Address of sender
    2. Address of receiver
    3. Amount in eth
    4. Keyword
    5. Message
    6. Timestamp

<u>Implementation</u>

Since the users are the main target group of our software, we will only be concerned about some important functions for the user. The user can send Ethereum by typing the account's address of the receiver.

Sending Ethereum: Sending Ethereum is the most important part of our system. We will describe this process in detail:-
First, we need to connect his wallet to our website. He/she can do it by clicking the connect wallet button. This will immediately trigger metamask connection, which is going to choose the account which we want to connect. After choosing the account click on next and click on connect. Instantly the address of your account is visible on the Ethereum card. Now write the address of the account where you want to send Ethereum. After that, write the amount of Ethereum you want to send. Now you can write a specific keyword that you want to attach with your transaction. This keyword is saved as data in the blockchain. Now you can also pass additional pieces of data or messages in the message box. Now click on the send button. Metamask will ask you to confirm the transaction. By Clicking on the confirm button the transaction has been executed successfully. The Ethereum will be visible on the receiver wallet. Users can also check the latest transaction details by scrolling down to the application.

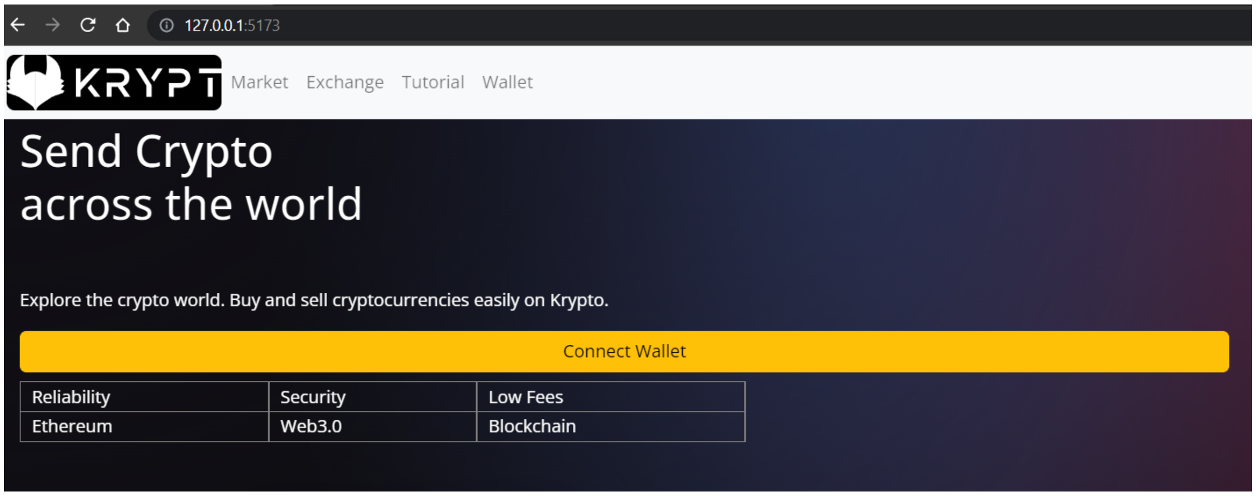
Using the follow application :

A person will be able to :

- Connect his crypto wallet using metamask.
- The address of the wallet has been visible to the Ethereum card.
- Check the amount of ether in his wallet.
- Able to send Ethereum to different wallets.
- Able to send GIFs with the ether as a gift.
- Able to send messages with ether.
- Able to check all the Transaction details.
- Able to interact with solidity Ethereum with smart contracts.

Result

1. Authenticated user connects to the wallet

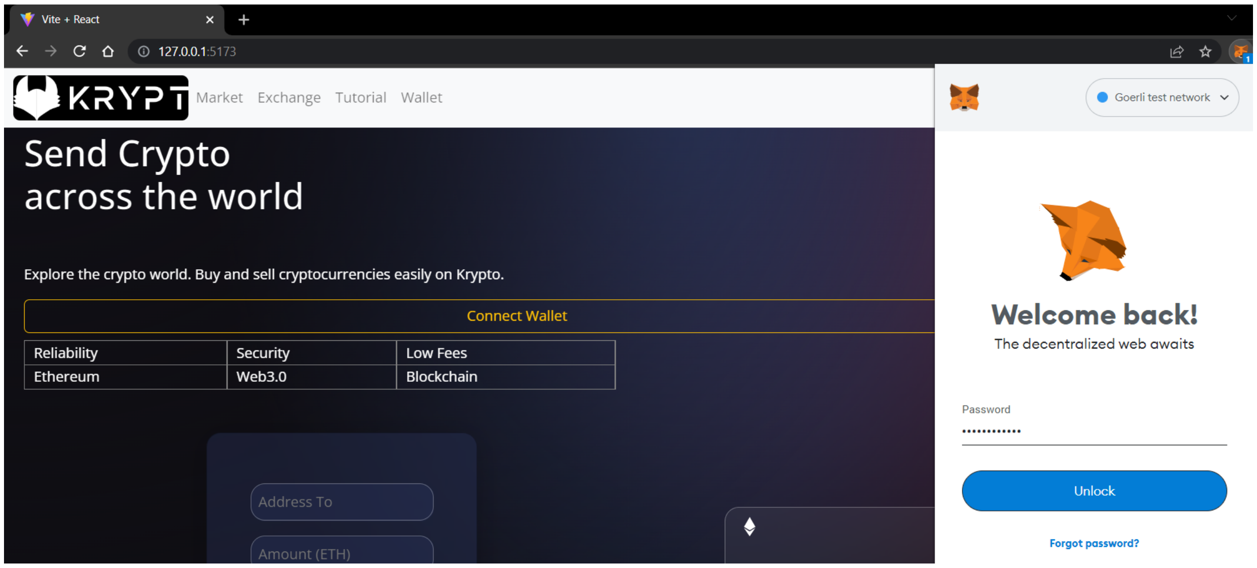


Send Crypto
across the world

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

Reliability	Security	Low Fees
Ethereum	Web3.0	Blockchain

2. Metamask notifies, once a user enters his/her credentials.



Send Crypto
across the world

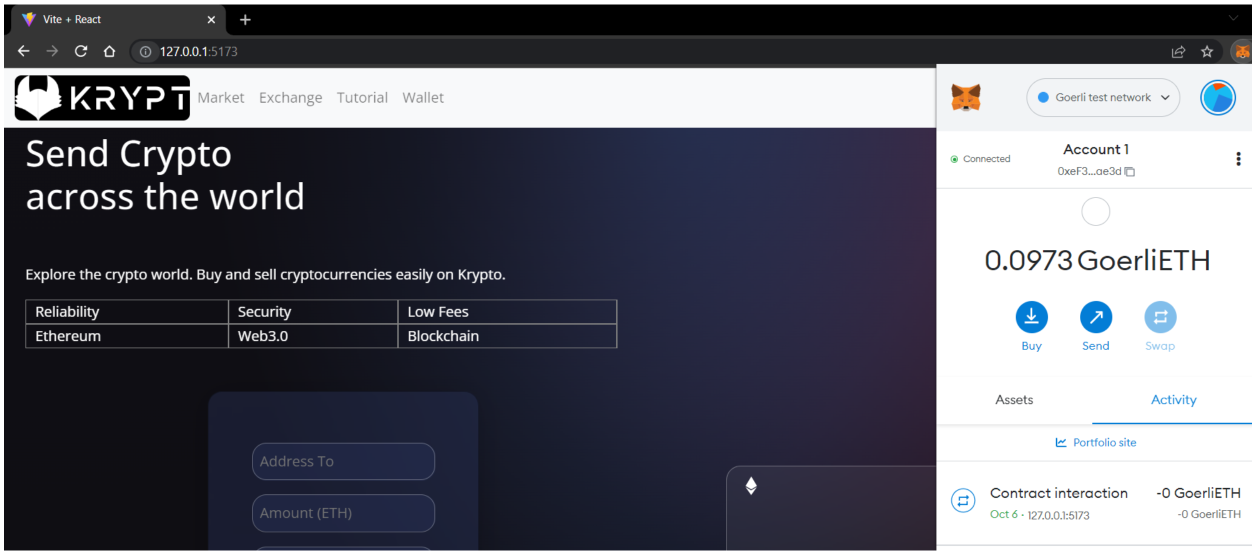
Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto.

Reliability	Security	Low Fees
Ethereum	Web3.0	Blockchain

Welcome back!
The decentralized web awaits

Password: ••••••••••••

3. Metamask account gets activated



Send Crypto
across the world

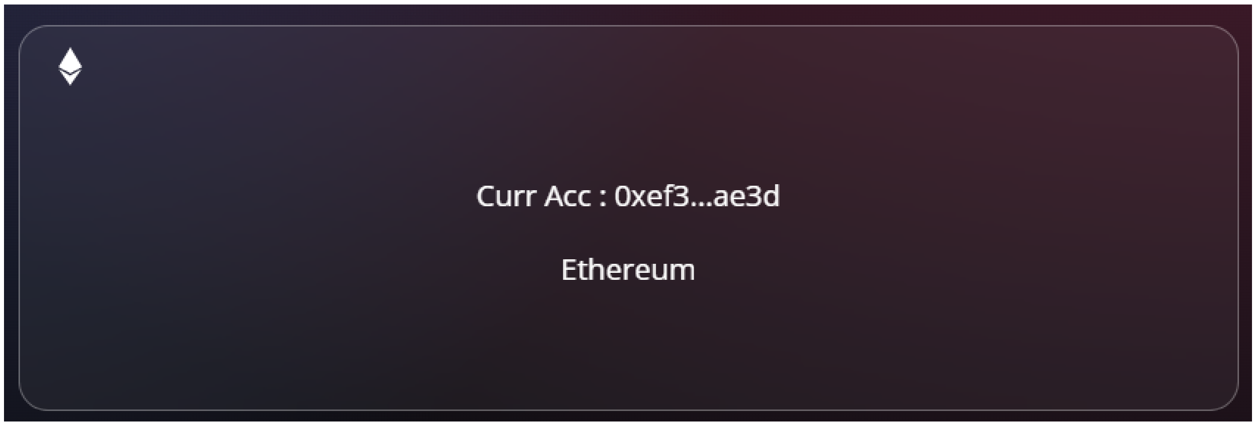
Explore the crypto world. Buy and sell cryptocurrencies easily on Krypto.

Reliability	Security	Low Fees
Ethereum	Web3.0	Blockchain

0.0973 GoerliETH

Contract interaction -0 GoerliETH
Oct 6 · 127.0.0.1:5173 -0 GoerliETH

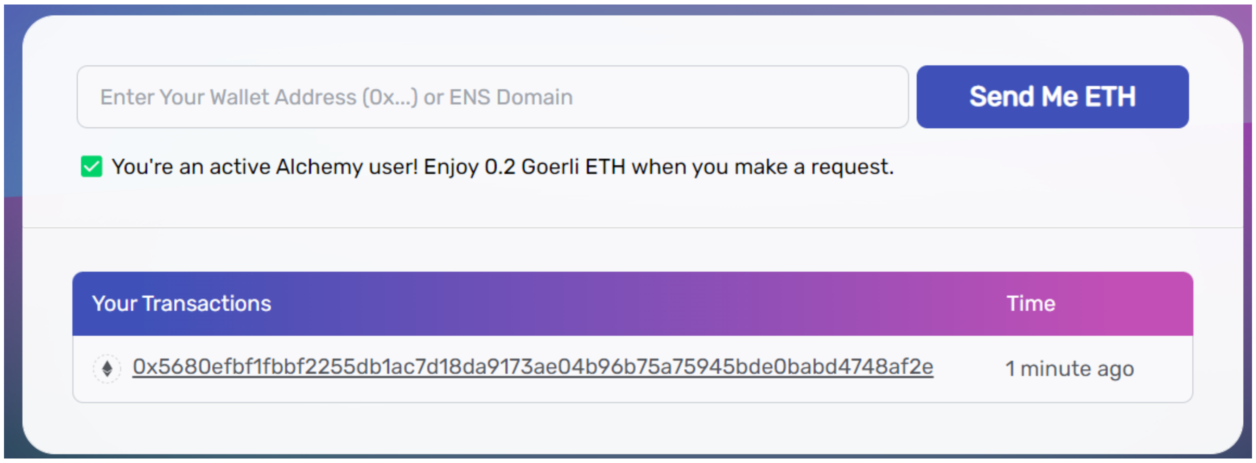
4. **Ethereum card displays the current account pf metamask user would use to perform a transaction.**



Curr Acc : 0xef3...ae3d

Ethereum

5. **Inorder to perform transactions on the test network we use Goerli test network and request to donate faucet Goerli eth on the pasted address.**

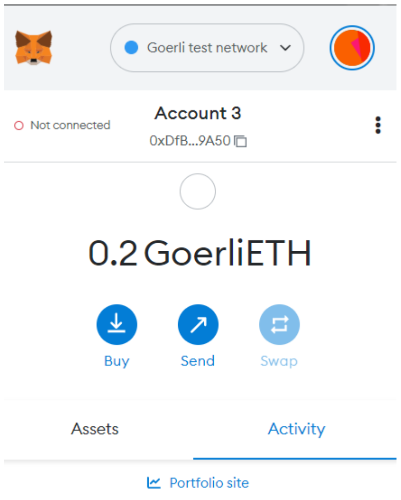

Enter Your Wallet Address (0x...) or ENS Domain

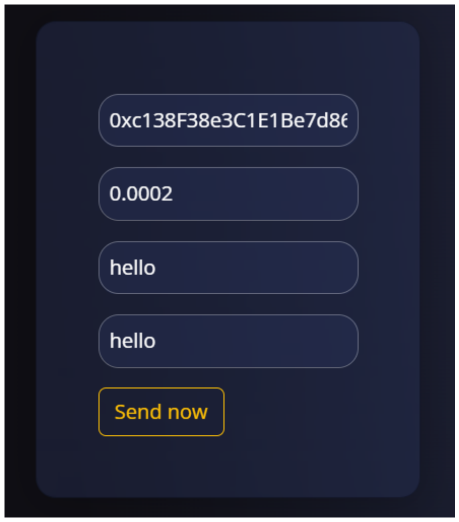
Send Me ETH

✓ You're an active Alchemy user! Enjoy 0.2 Goerli ETH when you make a request.

Your Transactions	Time
0x5680efbf1fbbf2255db1ac7d18da9173ae04b96b75a75945bde0babd4748af2e	1 minute ago

6. **Requested faucet is successfully sent and on the form towards the right we paste the address on which we want to send eth with the amount and additional message and keyword attached to a gif.**




0xc138F38e3C1E1Be7d8

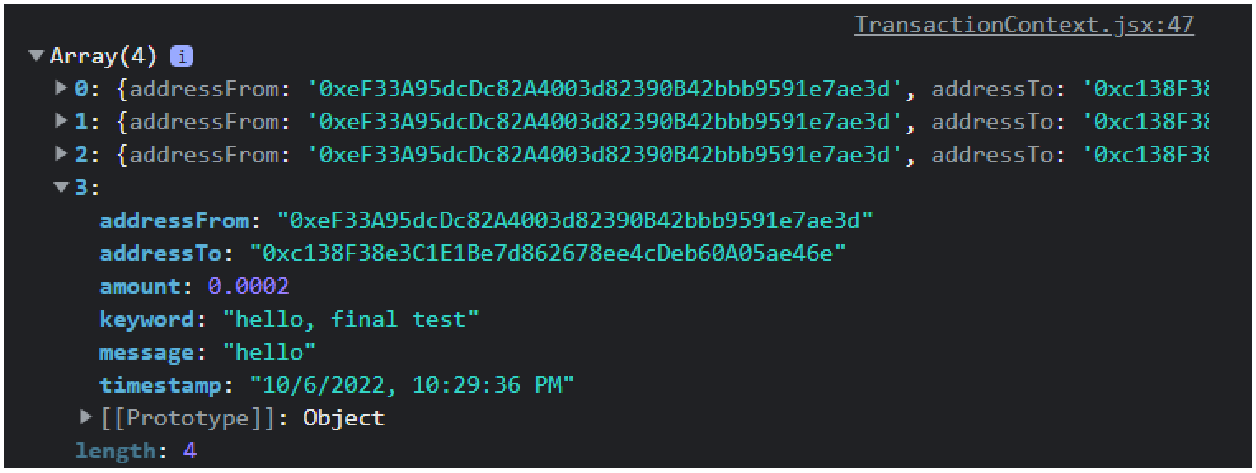
0.0002

hello

hello

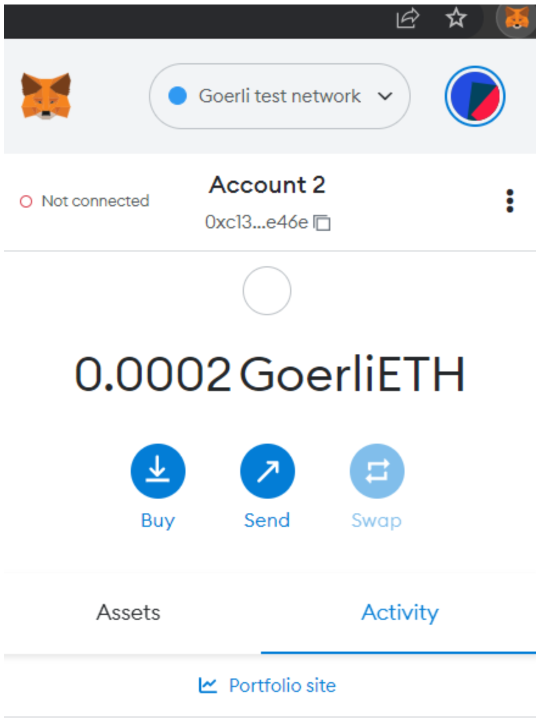
Send now

7. **An array of objects containing the transaction data is consoled.**



```
                                                              TransactionContext.jsx:47
▼Array(4) i
  ▶0: {addressFrom: '0xeF33A95dcDc82A4003d82390B42bbb9591e7ae3d', addressTo: '0xc138F38
  ▶1: {addressFrom: '0xeF33A95dcDc82A4003d82390B42bbb9591e7ae3d', addressTo: '0xc138F38
  ▶2: {addressFrom: '0xeF33A95dcDc82A4003d82390B42bbb9591e7ae3d', addressTo: '0xc138F38
  ▼3:
     addressFrom: "0xeF33A95dcDc82A4003d82390B42bbb9591e7ae3d"
     addressTo: "0xc138F38e3C1E1Be7d862678ee4cDeb60A05ae46e"
     amount: 0.0002
     keyword: "hello, final test"
     message: "hello"
     timestamp: "10/6/2022, 10:29:36 PM"
   ▶[[Prototype]]: Object
  length: 4
```

8. **Account2 with address 0xC138….ae46e which had 0 eth has now been transferred with 0.0002 goerli eth**
9. **Transaction cards are generated on each unique transaction with address from, to, amount, message and a gif (keyword).**



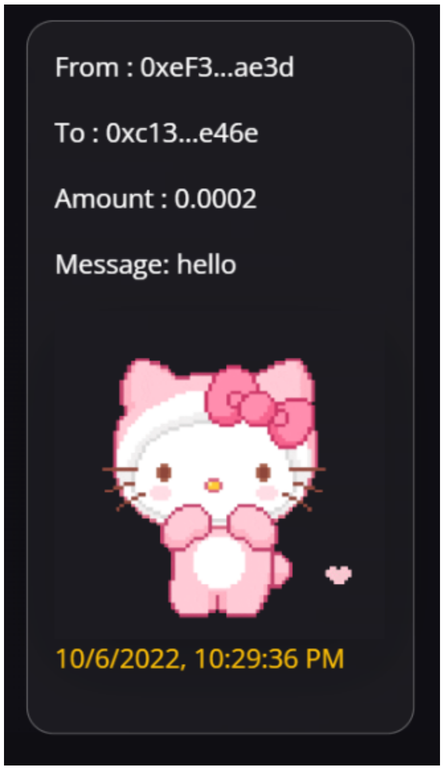
Goerli test network

Not connected

Account 2

0xc13...e46e

0.0002 GoerliETH



From : 0xeF3...ae3d

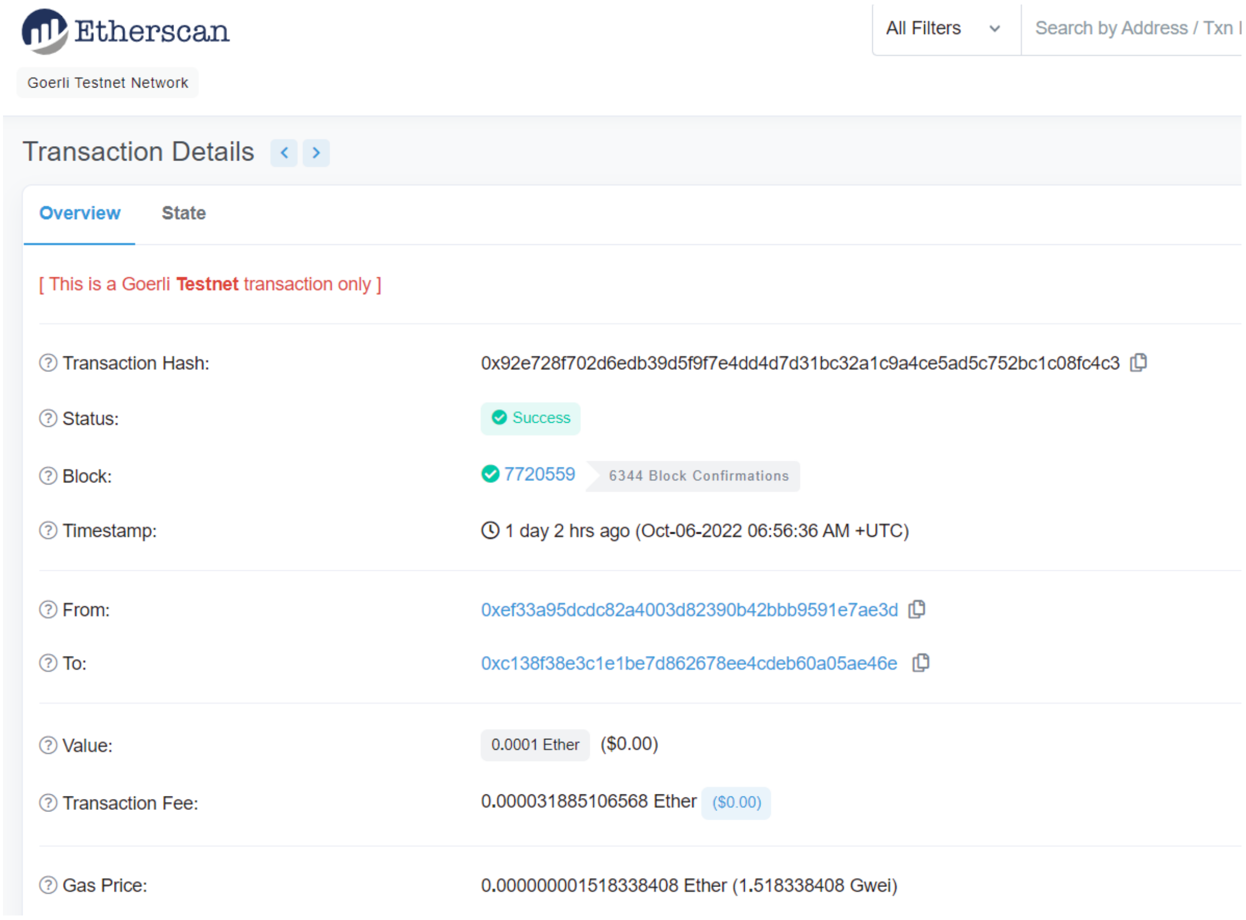
To : 0xc13...e46e

Amount : 0.0002

Message: hello

10/6/2022, 10:29:36 PM

10. **Onclick of the transaction card, etherscan of your address (transaction) is displayed on the blockchain network with the details attached on the screenshot.**



Transaction Details

[This is a Goerli **Testnet** transaction only]

Transaction Hash:	0x92e728f702d6edb39d5f9f7e4dd4d7d31bc32a1c9a4ce5ad5c752bc1c08fc4c3
Status:	Success
Block:	7720559 6344 Block Confirmations
Timestamp:	1 day 2 hrs ago (Oct-06-2022 06:56:36 AM +UTC)
From:	0xef33a95dcdc82a4003d82390b42bbb9591e7ae3d
To:	0xc138f38e3c1e1be7d862678ee4cdeb60a05ae46e
Value:	0.0001 Ether ($0.00)
Transaction Fee:	0.000031885106568 Ether ($0.00)
Gas Price:	0.000000001518338408 Ether (1.518338408 Gwei)

Technologies used

1.	Operating System:	Windows 11
2.	Programming language:	Solidity code
3.	Smart Contracts	
4.	Cryptocurrency Wallet:	Metamask pairing (Ethereum)
5.	Network:	Goerli test network
6.	Front-End:	HTML, CSS, React.js, Tailwind CSS, Bootstrap

References

a. Goerli faucet - https://goerlifaucet.com/
b. Alchemy - https://dashboard.alchemy.com/
c. Etherscan - https://goerli.etherscan.io/tx/{user_acc_hash}
d. Hardhat - https://hardhat.com/
[dependencies req: @nomiclabs/hardhat-waffle ethereum-waffle chai @nomiclabs/hardhat-ethers ethers]
e. Giphy - https://developers.giphy.com/docs/api/endpoint/#get-gif-by-id

Conclusion

In this project we have implemented a web3.0 application that allows users to make ethereum transactions through the blockchain network. The application is designed using Reactjs, in the frontend. This framework is connected to blockchain technology to perform decentralized storage of transactions. Krypt Web3.0 will be then paired with metamask,a cryptocurrency wallet that enables users to store Ether and other ERC-20 tokens. We use goerli ethereum test network in order to perform dummy transactions.
Using the follow application :

A person will be able to :

• Connect his crypto wallet using metamask.
• The address of the wallet has been visible to the Ethereum card.
• Check amount of ether in his wallet.
• Able to send Ethereum to different wallet.
• Able to send GIF with the ether as a gift.
• Able to send messages with ether.
• Able to check all the Transaction details.
• Able to interact with solidity Ethereum with smart contracts.

The project was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project. 1. We can provide users with the Customer Support System.
2. We can also create a sign in, sign up, sign out function into our system.
3. Can also make Customers Post their View on Websites.
4. The System has adequate scope for modification in future if it is necessary.
5. In the future we are going to add a crypto market into our system.
6. We will also add a crypto exchange feature into our system.
7. In future users will be able to send different crypto's from their account.