Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Practical No : 01

Title : Installation of metamask & study spending Ether per transaction.

objective : Students should be able to learn new technology such as metamask. Its application & implementation.

Prerequisite :
1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. working of blockchain.

Introduction :
- Blockchain can be described as a data structure that holds transactional records & while ensuring security, transperency, & decentralization

- A blockchain is distributed ledger that is completely open to any & everyone on the network. once an information is stored on a blockchain. It is extremly difficult to change it.

- Each transaction on a blockchain is secured with a digital signature that proves its authenticity.

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Blockchain technology allows all the network to participants to reach an agreement, commonly known as consesus.

Blockchain features

1. Decentralized : Blockchain are decentralized in nature meaning that no single person or group holds the authority of overall network.

2. Peer - to - Peer : The interaction betn two parties through a peer to peer model is easily accomplished without the requirement of any third party.

3. Immutable : The immutability property of a blockchain refers to the fact that any data once written on the blockchain cannot be changed.

4. Tamper - Proof : Blockchain are considered tamper proof as any change in even one single block can be detected & addressed smoothly.

Applications

1. Smart property
2. Energy
3. Healthcare
4. Distributed cloud storage
5. Machine learning
6. Digital Identity

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Benefits

1. Time saving : No central authority verification needed for settlements making the process faster and cheaper.
2. cost saving : No need for third party verification. Intermediaries & transaction efforts are reduced.
3. Tighter security : No one can temper with block-chain Data as it is shared among millions of participants.

How to use Metamask : A step by step guide

- step 1 - Install MetaMask on your browser
click on Install MetaMask as a google chrome extension.
click Add to chrome
click Add Extension.

- step 2 - Create an Account
click on the extension icon in the upper right corner to open MetaMask.
To install the latest version & be up to date, try it now.
click continue
you will be promted to create a new password
click create.

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

- Step 3 : Depositing funds
  click on view Accounts.

Advantages :

1. popular : It is commonly used so user only need one plugin to access a wide range of dapps.

2. simple : Instead of managing private keys users just needs to rember a list of words & transaction are signed on their behalf.

3. saves spaces : Users don't have to download the Ethereum blockchain, as metamask sends request to nodes outsides of the user's computer.

4. Integrated : Dapps are designed to work with Metamask, so it becomes much easier to send Ether in and out.

conclusion :

In this way we have explored concept Blockchain and metamat wallet for transaction of digital currency.

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Practical No :- 02

Title : create your own wallet using metamask for crypto transactions.

objectives : students should be able to learn about cryptocurrencies & learn how transaction done by using different digital currency.

Prerequisite : 1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept.
3. working of blockchain.

Introduction :
- cryptocurrency is a digital payment system that doesn't rely on banks to verify transaction. Its peer-to peer system that can enable anyone anywhere to send & receive payments.
- cryptocurrency received its name because it uses encryption to verify transaction. The aim of encryption to provide security & safety.

how does cryptocurrency work ?
cryptocurrencies run on distributed public ledgers called blockchain, a record of all transactions updated & held by currecy holders.

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

- If you own cryptocurrency, you don't own anything tangible. What you own is a key that allows you to move a record or a unit of measures from one person to another without a trusted third party.
- Although Bitcoin has been around since 2009 crypto currencies & applications of blockchain technology are still emerging in financial terms & more uses are expected in the future. Transaction including bonds, stocks & other financial assets could eventually be traded using the technology.

cryptocurrency examples

1. Bitcoin : founded in 2009, Bitcoin was the first cryptocurrency Developed by satoshi Nakamoto - widely believed to be a pseudonym for an individual or group of people whose precise identity remains unknown.
2. Ethereum : Developed in 2015. Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum.
3. Litecoin : This currency is most similar to bitcoin but moved more quickly to develop new innovations including faster payments & processes to allow more transactions.

4. Ripple : Ripple is distributed ledger system that was founded in 2012. Ripple can be used to track different kinds of transactions, not just cryptocurrency.

How to store cryptocurrency :

- once you have purchased cryptocurrency you need to store in safely to protect it from hacks or theft usually cryptocurrency stored in crypto wallets, which are physical devices or online software used store the private keys to your cryptocurrency securely.

- Hot wallet storage : 'hot wallet' refer to crypto storage that uses online s/w to protect the private keys to your assets.

- cold wallet storage : unlike hot wallets, cold wallets rely on offline electronic devices to securely store your private keys.

conclusion :

In this way we have explored concept crypto currency & learn how transactions are done using digital currency.

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Practical NO : 03

Title : Write a smart contract on test network, for Bank account of a customer for following operations :-
Deposite money
Withdraw money
Show balance

objective : student should be able to learn new technology such as metamask . Its application & implementation.

Prerequisite : 1. Basic knowledge of cryptocurrency
2. Basic knowledge of distributed computing concept
3. working of blockchain.

Theory :
The contract will allow deposits from any account & can be trusted to allow withdraws only by accounts that have sufficient funds to cover the requested withdrawal.

The post assumes that you are comfortable with the ether handling concepts introduced in our post writting a contract that handles Ether.

That post demonstrated how to restrict ether withdrawal to an owners account. It did this by persistently storing the owner account address & then comparing it to the msg sender value for any withdrawal attempt

I am going to generalize this contract to keep track of ether deposits based on the account address of the depositer. & then only allow that same account to make withdrawals of that Ether To do this we need a keep track of account balances.

For each depositing amount - a mapping from accounts to balances.

which will make this book keeping job quite simple. Here's the code to accept deposits & track account balances.

```
Program solidity ^ 0.4.19
contract Bank
mapping (address => uint 256) public balance of ;
function deposite (uint 256 amount) public payable {
require (msg.value == = amount) ;
balance of [msg.sender] + = amount ;
 }
 }
```

It's important to note that balance of keeps track of

the ether balances assigned to each account, but it does not actually move any ether anywhere. The bank contracts ether balance is the sum of all the balance of all accounts - only balance of tracks how much of that is assigned to each account.

Note also that this contract doesn't need a constructor there is no persistent state to initialize other than the balance of mapping, which already provides default values of 0.

Given the balance of mapping from account addresses to ether amounts the remaining code for a fully funal bank contract is pretty small I'll simply add a withdrawal function:

In the withdraw() funⁿ above it is very important to adjust balance of [msg.sender] before transferring ether to avoid an exploitable valnerability. The reason is specific to smart contract & the fact that a transfer to smart contract executes code in that smart contract.

Now, suppose that the code in withdraw() did not adjust balance of before making the transfer & suppose that msg sender was a malicious smart contract

Upon receiving the transfer - handled by msg sender's fallback fun^n that malicious contract could initiate another withdrawal from the banking contract. when the banking contract handles this second withdrawal request, it would have already transfered ether for the original withdrawal, but it would not updated balance, so it would allow this second withdrawal.

This vulenerability is called a reentracy bug because it happens when a smart contract invoke code in a different smart contract that then calls back into the original thereby reentering the exploitable contract.

To avoid this sort of reentray bug, follow the 'check - effects - Interactions pattern as described in the solidity documentation. The withdraw() fun^n above is an example of implementing this pattern

Shri Gajanan Maharaj Shikshan Prasarak Mandal's
SHARADCHANDRA PAWAR COLLEGE OF ENGINEERING
Otur (Dumbarwadi), Tal. Junnar, Dist. Pune - 412409

Practical No. : 04

Title : Write a program in solidity to create student data. use the following constructs :
- structures
- Arrays
- fallback

Deploy this as smart contract on Ethereum & observer the transaction fee & Gas values

objective : students should be able to learn about solidity. Its datatypes & implementation.

Prerequisite : 1. Basic programming logic
2. Basic knowledge of solidity

Theory :

Solidity - Arrays :

Arrays are data structures that store the fixed collection of elements of the same data types in which each & every element has a specific location called index. Instead of creating numerous individual variables of same type, we just declare one array of required size & store the elements in array & can be accessed using the index

creating an Array : synax :

<data type> <array name> [size] = <initialization>

Fixed size Array : The size of array should be predefined. The total number of elements should not exceed the size of array.

Dynamic Array : The size of array is not predefined when it is declared. As the element are added the size of array changes & at the runtime, the size of the array will be determined.

Array operations :

1. Accessing Array elements : The elements of array are accessed by using the index. If you want to access i^{th} element then you have to access (i-1) the index

2. PUSH : push is used when a new element is to be added in a dynamic array. The new element is always added at the last position of array.

3. POP : pop is used when the last element of array is to be removed in any dynamic array.

Solidity - structures :

structs in solidity allows you to create more complicate data types that have multiple properties. You can define your own type by creating a struct.

syntax : struct <structure_name> {
<data type> variable_1;
<data type> variable_2; }

solidity fallback :
The solidity fallback funⁿ is executed if none of the other funⁿ match the funⁿ identifier or no data was provided with the funⁿ call.

Properties of fallback funⁿ :
- Has no name or arguments
- cannot return anything
- can be defined once per contract
- It is mandatory to mark it external
- It also executed if the caller meant to call a funⁿ that is not available.

Conclusion :
In this way we have created array, structure & used fallback funⁿ in solidity.