Practical 6

Aim: Demonstrate cryptocurrency transaction processing

Theory:

Cryptocurrencies run on a distributed public ledger called blockchain, a record of all transactions updated and held by currency holders.

Units of cryptocurrency are created through a process called mining, which involves using computer power to solve complicated mathematical problems that generate coins. Users can also buy the currencies from brokers, then store and spend them using cryptographic wallets.

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 measure from one person to another without a trusted third party.

Although Bitcoin has been around since 2009, cryptocurrencies and applications of blockchain technology are still emerging in financial terms, and more uses are expected in the future. Transactions including bonds, stocks, and other financial assets could eventually be traded using the technology.

Transaction speed is one of the prime factors in Blockchains. Cryptocurrency having higher transaction speed results in the **most efficient cryptocurrency**. This means the higher the transaction speed of the Blockchains, the better its ability is to transfer data from one party to the other and confirm transactions. Here it is important to note that its transaction speed depends upon several factors such as block time, block size, transaction fees, and network traffic.

Crypto transaction speed can take a hit if the blockchain you transact has high network congestion, huge volume, and increased transaction fees.

Cryptocurrency with high transaction speed is crucial, and to maintain the speed, some specific factors must be considered.

Code:

```
const SHA26 = require('crypto-js/sha256')
class Block{
constructor(index , timestapms , data , previousHash = ''){
this.index = index;
this.timestapms = timestapms;
this.data = data;
this.previousHash = previousHash;
this.hash = this.calculateHash();
this.nounce = 0;
}
calculateHash(){
return SHA26(this.index + this.previousHash + this.timestapms +
JSON.stringify(this.data) + this.nounce).toString();}
mineBlock(difficulty){
while(this.hash.substring(0,difficulty) !== Array(difficulty + 1).join("0")){
this.nounce++;
this.hash = this.calculateHash();
}
console.log("Block mined :" + this.hash);
}
}
class Blockchain{
constructor(){
this.chain = [this.createGenesisisBlock()];
this.difficulty = 4;
}
createGenesisisBlock(){
return new Block(0, "09/09/2022", "Genesis Block","0");
}
getLatestBlock(){
return this.chain[this.chain.length - 1]
}
addBlock(newBlock){
newBlock.previousHash = this.getLatestBlock().hash;
newBlock.mineBlock(this.difficulty);
this.chain.push(newBlock);
}
}
let ZeeCoin = new Blockchain();
console.log("mining block 1..");
ZeeCoin.addBlock(new Block(1,"18/09/2021",{amount : 4}));
console.log("mining block 2..");
ZeeCoin.addBlock(new Block(2,"28/09/2021",{amount : 8}));
```

Output:

```
Zeeshan's PC@MSI MINGW64 /d/SEM-3 practicals/Blockchain-practicals
$ node "d:\SEM-3 practicals\Blockchain-practicals\index.js"
mining block 1..
Block mined :0000d974650e46866e99f6564901dcfd30b7d1720bec5dd501f5c58f8c079132
mining block 2..
Block mined :0000b37f9b4cddecf2666bdd5185b06218a5ab01f9e9389f4daed817b886b493
```

Conclusion:

Hence, we successfully implemented cryptocurrency transaction processing.