PRACTICAL 5

AIM:

Study and Implement Block structure and perform following tasks using Node js/Python/Java.

a. Create Genesis Block and display.
b. Create three Blocks and link all of them with Genesis block in chronological order.
c. Perform transactions by sending and receiving amount.
d. Mine the transaction (Validation/Verification) and add into block.
e. After adding Block into chain, try to modify or delete the block.
f. View the transaction and Blocks creation over the network.

PREREQUISITES:

Node.js installed on your machine.

A code editor, such as Visual Studio Code, Sublime Text, or any other.

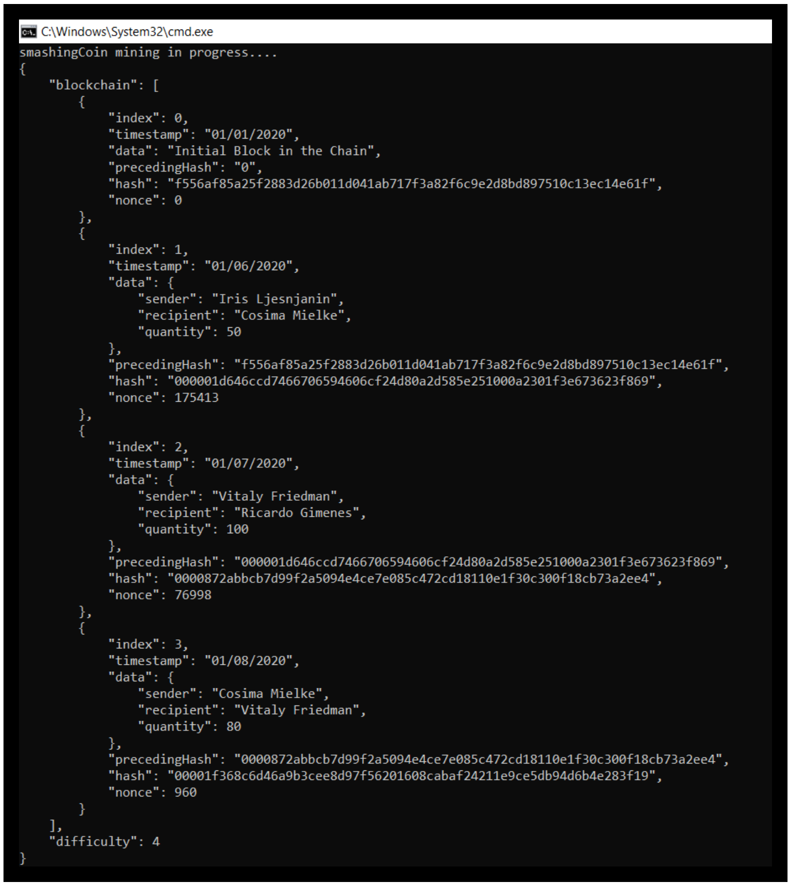
CODE:

```
const SHA256 = require("crypto-js/sha256");
class CryptoBlock {
  constructor(index, timestamp, data, precedingHash = " ") {
    this.index = index;
    this.timestamp = timestamp;
    this.data = data;
    this.precedingHash = precedingHash;
    this.hash = this.computeHash();
    this.nonce = 0;
  }
  computeHash() {
    return SHA256(
      this.index +
        this.precedingHash +
        this.timestamp +
        JSON.stringify(this.data) +
        this.nonce
    ).toString();
  }
  proofOfWork(difficulty) {
    while (
      this.hash.substring(0, difficulty) !== Array(difficulty + 1).join("0")
    ) {
      this.nonce++;
      this.hash = this.computeHash();
    }
  }
}
class CryptoBlockchain {
  constructor() {
    this.blockchain = [this.startGenesisBlock()];
    this.difficulty = 4;
  }
  startGenesisBlock() {
    return new CryptoBlock(0, "01/01/2020", "Initial Block in the Chain", "0");
  }
  obtainLatestBlock() {
    return this.blockchain[this.blockchain.length - 1];
  }
  addNewBlock(newBlock) {
    newBlock.precedingHash = this.obtainLatestBlock().hash;
    newBlock.proofOfWork(this.difficulty);
    this.blockchain.push(newBlock);
  }
  checkChainValidity() {
    for (let i = 1; i < this.blockchain.length; i++) {
      const currentBlock = this.blockchain[i];
      const precedingBlock = this.blockchain[i - 1];

      if (currentBlock.hash !== currentBlock.computeHash()) {
        return false;
      }
      if (currentBlock.precedingHash !== precedingBlock.hash) return false;
    }
    return true;
  }
}
let smashingCoin = new CryptoBlockchain();
console.log("smashingCoin mining in progress....");
smashingCoin.addNewBlock(
  new CryptoBlock(1, "01/06/2020", {
    sender: "Iris Ljesnjanin",
    recipient: "Cosima Mielke",
    quantity: 50
  })
);
smashingCoin.addNewBlock(
  new CryptoBlock(2, "01/07/2020", {
    sender: "Vitaly Friedman",
    recipient: "Ricardo Gimenes",
    quantity: 100
  })
);
smashingCoin.addNewBlock(
    new CryptoBlock(3, "01/08/2020", {
      sender: "Cosima Mielke",
      recipient: "Vitaly Friedman",
      quantity: 80
    })
  );

console.log(JSON.stringify(smashingCoin, null, 4));
```

OUTPUT:



```
C:\Windows\System32\cmd.exe
smashingCoin mining in progress....
{
    "blockchain": [
        {
            "index": 0,
            "timestamp": "01/01/2020",
            "data": "Initial Block in the Chain",
            "precedingHash": "0",
            "hash": "f556af85a25f2883d26b011d041ab717f3a82f6c9e2d8bd897510c13ec14e61f",
            "nonce": 0
        },
        {
            "index": 1,
            "timestamp": "01/06/2020",
            "data": {
                "sender": "Iris Ljesnjanin",
                "recipient": "Cosima Mielke",
                "quantity": 50
            },
            "precedingHash": "f556af85a25f2883d26b011d041ab717f3a82f6c9e2d8bd897510c13ec14e61f",
            "hash": "000001d646ccd7466706594606cf24d80a2d585e251000a2301f3e673623f869",
            "nonce": 175413
        },
        {
            "index": 2,
            "timestamp": "01/07/2020",
            "data": {
                "sender": "Vitaly Friedman",
                "recipient": "Ricardo Gimenes",
                "quantity": 100
            },
            "precedingHash": "000001d646ccd7466706594606cf24d80a2d585e251000a2301f3e673623f869",
            "hash": "0000872abbcb7d99f2a5094e4ce7e085c472cd18110e1f30c300f18cb73a2ee4",
            "nonce": 76998
        },
        {
            "index": 3,
            "timestamp": "01/08/2020",
            "data": {
                "sender": "Cosima Mielke",
                "recipient": "Vitaly Friedman",
                "quantity": 80
            },
            "precedingHash": "0000872abbcb7d99f2a5094e4ce7e085c472cd18110e1f30c300f18cb73a2ee4",
            "hash": "00001f368c6d46a9b3cee8d97f56201608cabaf24211e9ce5db94d6b4e283f19",
            "nonce": 960
        }
    ],
    "difficulty": 4
}
```

CONCLUSION:

In this practical, we thoroughly understood the concept of blockchain and implemented our own cryptocurrency using Node JS with proof of work consensus mechanism.