19BCE072

BCT Pract 2

Gondaliya krish dhirubhai

**CODE:**

const crypto = require("crypto"),

SHA256 = (message) =>

crypto.createHash("sha256").update(message).digest("hex");

class Block {

constructor(index, timestamp, data, prevHash = "") {

this.index = index;

this.timestamp = timestamp;

this.data = data;

this.hash = this.getHash();

this.prevHash = prevHash;

}

getHash() {

return SHA256(

this.index +

this.prevHash +

this.timestamp +

JSON.stringify(this.data)

).toString();

}

}

class BlockChain {

constructor() {

this.chain = [];

}

getLastBlock() {

return this.chain[this.chain.length - 1];

}

genesisBlock(index, data) {

let date = Date.now().toString();

let temp\_block = new Block(index, date, data);

this.chain.push(temp\_block);

}

addBlock(index, data) {

let date = Date.now().toString();

let block = new Block(index, date, data);

block.prevHash = this.getLastBlock().hash;

this.chain.push(block);

}

changeBlock(index) {

let tamp\_date = Date.now().toString();

let tamp\_hash = SHA256(index + JSON.stringify(tamp\_date)).toString();

this.chain[index].hash = tamp\_hash;

// this.chain.push(block);

}

changeBlockToOriginal(index) {

let orig\_hash = this.chain[index + 1].prevHash;

this.chain[index].hash = orig\_hash;

}

}

let blockchain = new BlockChain();

blockchain.genesisBlock(0, "Genesis Block");

for (let i = 1; i < 10; i++) {

let a = "Block No. " + i;

blockchain.addBlock(i, a);

}

console.log(blockchain.chain);

function validation(b) {

var k = false;

for (let i = 1; i < b.chain.length; i++) {

if (b.chain[i].prevHash != b.chain[i - 1].hash) {

console.log("Data tampered at Block no. " + `${i - 1}`);

k = true;

}

}

if (!k) {

console.log("No data Tampered!");

}

}

setInterval(() => {

validation(blockchain);

}, 1000);

setTimeout(() => {

blockchain.changeBlock(2);

}, 3000);

setTimeout(() => {

blockchain.changeBlockToOriginal(2);

}, 6000);



