Practical No. 10

Name: Bhairavi Narendra Rewatkar

Roll No.: DMET1221006

Subject: Blockchain Technology Laboratory

Date: 13/01/2025

Title: Integration of the Blockchain.

Aim: Write a program to implement a method to verify the integrity of the Blockchain.

Source Code:

HashUtil.java

```
import java.security.MessageDigest;
public class HashUtil {
    public static String applySHA256(String input) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hashBytes = digest.digest(input.getBytes("UTF-8"));
            StringBuilder hexString = new StringBuilder();
            for (byte b : hashBytes) {
                String hex = Integer.toHexString(0xff & b);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (Exception e) {
            throw new RuntimeException(e);   }    }}
```

Block.java

```
public class Block {
    private int index;
    private long timestamp;
    private String previousHash;
    private String currentHash;
    private int nonce;
    private String data;
```

```
    public Block(int index, String previousHash, String data) {
        this.index = index;
        this.previousHash = previousHash;
        this.data = data;
        this.timestamp = System.currentTimeMillis();
        this.nonce = 0;
        this.currentHash = calculateHash(); }
    public String calculateHash() {
        String content = index + Long.toString(timestamp) + previousHash + nonce + data;
        return HashUtil.applySHA256(content);   }
    public void mineBlock(int difficulty) {
        String target = new String(new char[difficulty]).replace('\0', '0'); // Difficulty target
        while (!currentHash.substring(0, difficulty).equals(target)) {
            nonce++;
            currentHash = calculateHash();}}
    public String getHash() {
        return currentHash;    }
    public String getPreviousHash() {
        return previousHash;    }
    @Override
    public String toString() {
        return "Block{" +
                "index=" + index +
                ", timestamp=" + timestamp +
                ", previousHash='" + previousHash + '\'' +
                ", currentHash='" + currentHash + '\'' +
                ", nonce=" + nonce +
                ", data='" + data + '\'' +
                '}';}}
```

Blockchain.java

```
import java.util.ArrayList;
import java.util.List;
```

```
public class Blockchain {
    private List<Block> chain;
    private int difficulty;
    public Blockchain(int difficulty) {
        this.chain = new ArrayList<>();
        this.difficulty = difficulty;
        chain.add(createGenesisBlock());  }
    private Block createGenesisBlock() {
        return new Block(0, "0", "Genesis Block");}
    public void addBlock(String data) {
        Block previousBlock = chain.get(chain.size() - 1);
        Block newBlock = new Block(chain.size(), previousBlock.getHash(), data);
        newBlock.mineBlock(difficulty);
        chain.add(newBlock);    }
    public boolean isChainValid() {
        for (int i = 1; i < chain.size(); i++) {
            Block currentBlock = chain.get(i);
            Block previousBlock = chain.get(i - 1);
            if (!currentBlock.getHash().equals(currentBlock.calculateHash())) {
                return false; // Current block hash is not valid
            }
            if (!currentBlock.getPreviousHash().equals(previousBlock.getHash())) {
                return false; // Previous block hash doesn't match}       }
        return true; // Blockchain is valid
    }
    public void printBlockchain() {
        for (Block block : chain) {
            System.out.println(block);}}
    public Block getLastBlock() {
        return chain.get(chain.size() - 1);  }}
```

Main.java

```
public class Main {
```

```
public static void main(String[] args) {
    Blockchain blockchain = new Blockchain(4); // Difficulty set to 4
    System.out.println("Mining block 1...");
    blockchain.addBlock("First Block after Genesis");
    System.out.println("Mining block 2...");
    blockchain.addBlock("Second Block after Genesis");
    System.out.println("Mining block 3...");
    blockchain.addBlock("Third Block after Genesis");
    blockchain.printBlockchain();
    // Verify blockchain integrity
    boolean isValid = blockchain.isChainValid();
    System.out.println("Is blockchain valid? " + isValid);
    // Tamper with the blockchain (for testing)
    System.out.println("Tampering with the blockchain...");
    blockchain.getLastBlock().toString(); // Simulate tampering by modifying data
    // Verify blockchain integrity after tampering
    isValid = blockchain.isChainValid();
    System.out.println("Is blockchain valid after tampering? " + isValid);}}
```

Output:

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>cd Desktop

C:\Users\STUDENT\Desktop>javac HashUtil.java Block.java Blockchain.java Main.java

C:\Users\STUDENT\Desktop>java Main.java
Mining block 1...
Mining block 2...
Mining block 3...
Block{index=0, timestamp=1736747500677, previousHash='0', currentHash='e7b9e6d78d5680febba966ecc134c9fea7364dfb8f4cb1cf6
012c27a106a3828', nonce=0, data='Genesis Block'}
Block{index=1, timestamp=1736747500734, previousHash='e7b9e6d78d5680febba966ecc134c9fea7364dfb8f4cb1cf6012c27a106a3828',
 currentHash='0000ed2f2148a4bd5233aa2ae8265309b82029cc2730715c769867c0655c2727', nonce=222052, data='First Block after G
enesis'}
Block{index=2, timestamp=1736747500958, previousHash='0000ed2f2148a4bd5233aa2ae8265309b82029cc2730715c769867c0655c2727',
 currentHash='000065ae9a3649263cc01af8a900148630ed4ce641843159f3c630f3f457ed44', nonce=60854, data='Second Block after G
enesis'}
Block{index=3, timestamp=1736747500996, previousHash='000065ae9a3649263cc01af8a900148630ed4ce641843159f3c630f3f457ed44',
 currentHash='0000b5330cb638515875ea9ec782f448e099bdaab6aa7d2c7b44a7e3790e0126', nonce=3685, data='Third Block after Gen
esis'}
Is blockchain valid? true
Tampering with the blockchain...
Is blockchain valid after tampering? true

C:\Users\STUDENT\Desktop>
```