

Practical No: 15

Name: Bhairavi Narendra Rewatkar

Roll No.: DMET1221006

Subject: Blockchain Technology Laboratory

Title: Decentralized Voting System

Aim: Write a program to implement a decentralized voting system using blockchain technology.

Source Code:

Vote.java

```
import java.security.MessageDigest;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
class Vote {
    public String voterId;
    public String candidate;
    public long timestamp;
    public String hash;
    public Vote(String voterId, String candidate, long timestamp) {
        this.voterId = voterId;
        this.candidate = candidate;
        this.timestamp = timestamp;
        this.hash = calculateHash();
    }
    public String calculateHash() {
        String input = voterId + candidate + timestamp;
        return applySHA256(input);
    }
    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 hashByte : hashBytes) {
                String hex = Integer.toHexString(0xff & hashByte);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Block.java

```
class Block {
    public String previousHash;
    public Vote vote;
```

```

public String hash;
public long timestamp;
// Block constructor
public Block(String previousHash, Vote vote, long timestamp) {
    this.previousHash = previousHash;
    this.vote = vote;
    this.timestamp = timestamp;
    this.hash = calculateHash();
}
public String calculateHash() {
    String input = previousHash + vote.hash + timestamp;
    return Vote.applySHA256(input);
}
}
Blockchain.java
class Blockchain {
    public List<Block> chain;
    public Blockchain() {
        chain = new ArrayList<>();
        chain.add(createGenesisBlock());
    }
    private Block createGenesisBlock() {
        return new Block("0", new Vote("Genesis", "None",
            currentTimestamp()), currentTimestamp());
    }
    public Block getLatestBlock() {
        return chain.get(chain.size() - 1);
    }
    public void addVote(Vote vote) {
        Block latestBlock = getLatestBlock();
        Block newBlock = new Block(latestBlock.hash, vote,
            currentTimestamp());
        chain.add(newBlock);
    }
    public boolean validateVotes() {
        for (int i = 1; i < chain.size(); i++) {
            Block currentBlock = chain.get(i);
            Block previousBlock = chain.get(i - 1);
            if
            (!currentBlock.hash.equals(currentBlock.calculateHash())) {
                System.out.println("Block " + i + " has been tampered
                with.");
                return false;
            }
            if
            (!currentBlock.previousHash.equals(previousBlock.hash)) {
                System.out.println("Block " + i + "'s previous hash
                doesn't match.");
                return false;
            }
        }
    }
}

```

```

return true;
}
public long currentTimestamp() {
return new Date().getTime();
}}
VotingSystem.java
public class VotingSystem {
public static void main(String[] args) {
Blockchain blockchain = new Blockchain();
Vote vote1 = new Vote("Voter1", "Alice",
blockchain.currentTimestamp());
blockchain.addVote(vote1);
Vote vote2 = new Vote("Voter2", "Bob",
blockchain.currentTimestamp());
blockchain.addVote(vote2);
System.out.println("Blockchain is valid: " +
blockchain.validateVotes());
for (Block block : blockchain.chain) {
System.out.println("Block [Previous Hash: " +
block.previousHash + ", Hash: " + block.hash +
", Vote: " + block.vote.voterId + " voted for " +
block.vote.candidate + ", Timestamp: " + block.timestamp + "]);
}
}
}
}

```

Output:

```

Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>cd Desktop
C:\Users\STUDENT\Desktop>javac Main.java
C:\Users\STUDENT\Desktop>Main.java
C:\Users\STUDENT\Desktop>java Main.java
Block mined: 00009b1ff393beb88bf3bd476aecf2477c6022d6ae5527c05bb4454b6cd89fe6
Block mined: 0000019cecdcea1a0cd45cc7df58ae7669dc7a8abdb8b81f4b555d5441e0c20d
Block mined: 00006a7beba3c1c987c9a5b25720a4989484f3c6a05a5bfc7047c3970b899b72
Blockchain is valid: true
Block 0 [Hash: 304477f412ca133cce41d88b07db148866d245e421e12aac77136319351eafdb, Previous Hash: 0, Timestamp: 1739166584135]
Block 1 [Hash: 00009b1ff393beb88bf3bd476aecf2477c6022d6ae5527c05bb4454b6cd89fe6, Previous Hash: 304477f412ca133cce41d88b07db148866d245e421e12aac77136319351eafdb, Timestamp: 1739166584161]
Block 2 [Hash: 0000019cecdcea1a0cd45cc7df58ae7669dc7a8abdb8b81f4b555d5441e0c20d, Previous Hash: 00009b1ff393beb88bf3bd476aecf2477c6022d6ae5527c05bb4454b6cd89fe6, Timestamp: 1739166584300]
Block 3 [Hash: 00006a7beba3c1c987c9a5b25720a4989484f3c6a05a5bfc7047c3970b899b72, Previous Hash: 0000019cecdcea1a0cd45cc7df58ae7669dc7a8abdb8b81f4b555d5441e0c20d, Timestamp: 1739166584400]

C:\Users\STUDENT\Desktop>|

```