**TREUE TECHNOLIGIES**

**INTERNSHIP PROGRAME**

**TASK -5**

**ONLINE VOTING SYSTEM :**

**Create an online voting system that enables users to participate in elections or polls securely and conveniently. The system should provide user registration and authentication, ballot creation, and vote casting functionalities. Implement features like candidate or option selection, vote counting, and result generation. Focus on ensuring the security and integrity of the voting process, incorporating measures such as encryption, access controls, and audit trails.**

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.util.\*;

class User {

private String username;

private String passwordHash;

public User(String username, String password) {

this.username = username;

this.passwordHash = hashPassword(password);

}

public boolean authenticate(String password) {

String inputHash = hashPassword(password);

return inputHash.equals(passwordHash);

}

private String hashPassword(String password) {

try {

MessageDigest md = MessageDigest.getInstance("SHA-256");

byte[] hashBytes = md.digest(password.getBytes());

StringBuilder hexHash = new StringBuilder();

for (byte b : hashBytes) {

hexHash.append(String.format("%02x", b));

}

return hexHash.toString();

} catch (NoSuchAlgorithmException e) {

throw new RuntimeException("Password hashing failed.");

}

}

public String getUsername() {

return username;

}

}

class Ballot {

private String title;

private List<String> options;

private Map<String, Integer> results;

public Ballot(String title, List<String> options) {

this.title = title;

this.options = options;

this.results = new HashMap<>();

for (String option : options) {

results.put(option, 0);

}

}

public String getTitle() {

return title;

}

public List<String> getOptions() {

return options;

}

public Map<String, Integer> getResults() {

return results;

}

public void castVote(String option) {

if (options.contains(option)) {

results.put(option, results.get(option) + 1);

}

}

}

class VotingSystem {

private List<User> users;

private List<Ballot> ballots;

public VotingSystem() {

this.users = new ArrayList<>();

this.ballots = new ArrayList<>();

}

public void registerUser(String username, String password) {

users.add(new User(username, password));

}

public User login(String username, String password) {

for (User user : users) {

if (user.getUsername().equals(username) && user.authenticate(password)) {

return user;

}

}

return null;

}

public void createBallot(String title, List<String> options) {

ballots.add(new Ballot(title, options));

}

public List<Ballot> getBallots() {

return ballots;

}

public void castVote(User user, Ballot ballot, String option) {

if (user != null && ballots.contains(ballot)) {

ballot.castVote(option);

}

}

}

public class OnlineVotingSystem {

public static void main(String[] args) {

VotingSystem votingSystem = new VotingSystem();

// Register users

votingSystem.registerUser("user1", "password1");

votingSystem.registerUser("user2", "password2");

// Create a ballot

List<String> options = Arrays.asList("Option A", "Option B", "Option C");

votingSystem.createBallot("Election 2023", options);

// Simulate user login

User user = votingSystem.login("user1", "password1");

// Cast a vote

if (user != null) {

Ballot ballot = votingSystem.getBallots().get(0);

votingSystem.castVote(user, ballot, "Option A");

System.out.println("Vote cast successfully.");

} else {

System.out.println("Login failed.");

}

// Get and display ballot results

Ballot ballot = votingSystem.getBallots().get(0);

System.out.println("Ballot Title: " + ballot.getTitle());

Map<String, Integer> results = ballot.getResults();

for (Map.Entry<String, Integer> entry : results.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue() + " votes");

}

}

}