<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Cybersecurity Simulation - T129</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #0f172a;

color: #f1f5f9;

padding: 20px;

}

h1, h2 {

color: #38bdf8;

}

button {

margin: 5px;

padding: 10px 20px;

border: none;

background-color: #1e40af;

color: white;

border-radius: 8px;

cursor: pointer;

}

input {

margin: 5px;

padding: 8px;

border-radius: 6px;

border: 1px solid #64748b;

width: 200px;

}

#output {

margin-top: 20px;

white-space: pre-line;

background-color: #1e293b;

padding: 15px;

border-radius: 10px;

max-height: 300px;

overflow-y: auto;

}

</style>

</head>

<body>

<h1>Cybersecurity Simulation - Team T129</h1>

<h2>Student Innovation Project</h2>

<div id="controls"></div>

<div id="input-area"></div>

<div id="output"></div>

<script>

let identityVerified = false;

let ledger = [];

const controls = document.getElementById("controls");

const inputArea = document.getElementById("input-area");

const output = document.getElementById("output");

const features = [

"Identity Check",

"Access Logs",

"Certificate Check",

"Voting Access",

"Threat Monitor",

"Data Integrity",

"Smart Contract Action",

"IoT Device Authentication",

"Blockchain Certificate Verification",

"View Blockchain Ledger",

];

features.forEach((label, index) => {

const btn = document.createElement("button");

btn.textContent = ${index + 1}. ${label};

btn.onclick = () => runCase(index + 1);

controls.appendChild(btn);

});

function print(msg) {

output.innerText += msg + "\n";

output.scrollTop = output.scrollHeight;

}

function runCase(choice) {

inputArea.innerHTML = "";

output.innerText = "";

if (choice === 1) {

inputArea.innerHTML = `

<input placeholder="Enter ID" id="uid" /><br>

<input type="password" placeholder="Password" id="pass" /><br>

<input type="password" placeholder="Confirm Password" id="confirm" /><br>

<button onclick="verifyIdentity()">Submit</button>

`;

} else if (!identityVerified) {

print("❌ Identity check required before accessing this feature.");

} else {

switch (choice) {

case 2:

print("Access Log Started...");

for (let i = 1; i <= 5; i++) {

let log = Access event ${i}: Module ${String.fromCharCode(64 + i)};

print("✅ " + log);

ledger.push(log);

}

break;

case 3:

inputArea.innerHTML = <input placeholder='Certificate ID' id='cert'><button onclick='checkCert()'>Validate</button>;

break;

case 4:

inputArea.innerHTML = <input placeholder='Voter ID' id='voter'><button onclick='castVote()'>Vote</button>;

break;

case 5:

print("Threat Monitoring...");

for (let i = 1; i <= 3; i++) {

let scan = ✔ Scan #${i}: No threats.;

print(scan);

ledger.push(scan);

}

break;

case 6:

inputArea.innerHTML = `

<input placeholder='Enter Hash' id='hash1'><br>

<input placeholder='Re-enter Hash' id='hash2'><br>

<button onclick='checkHash()'>Verify</button>

`;

break;

case 7:

[

"✔ Verified sender.",

"✔ Checked balance.",

"✔ Transfer executed.",

"✔ Transaction logged."

].forEach(action => {

print(action);

ledger.push(action);

});

break;

case 8:

inputArea.innerHTML = <input placeholder='MAC Address' id='mac'><button onclick='verifyMAC()'>Verify</button>;

break;

case 9:

inputArea.innerHTML = <input placeholder='Blockchain Cert ID' id='bcert'><button onclick='verifyBlockchain()'>Check</button>;

break;

case 10:

print("=== Blockchain Ledger Preview ===");

ledger.forEach((log, i) => print(Block ${i + 1}: ${log}));

break;

}

}

}

function verifyIdentity() {

let uid = document.getElementById("uid").value;

let pass = document.getElementById("pass").value;

let confirm = document.getElementById("confirm").value;

if (pass === confirm && pass.length >= 6) {

identityVerified = true;

print("✅ Identity Verified.");

ledger.push("Identity verified for user: " + uid);

} else {

identityVerified = false;

print("❌ Identity verification failed.");

ledger.push("Identity check failed for user: " + uid);

}

}

function checkCert() {

const cert = document.getElementById("cert").value;

if (cert.length === 10) {

print("✅ Certificate Valid.");

ledger.push("Certificate " + cert + " validated.");

} else {

print("❌ Invalid Certificate.");

ledger.push("Certificate validation failed: " + cert);

}

}

function castVote() {

const voter = document.getElementById("voter").value;

if (voter.length === 6) {

print("✅ Vote Encrypted.");

ledger.push("Vote cast by ID: " + voter);

} else {

print("❌ Invalid Voter ID.");

}

}

function checkHash() {

const h1 = document.getElementById("hash1").value;

const h2 = document.getElementById("hash2").value;

if (h1 === h2) {

print("✅ Data Integrity Confirmed.");

ledger.push("Hash match: " + h1);

} else {

print("❌ Data mismatch.");

ledger.push("Hash mismatch");

}

}

function verifyMAC() {

const mac = document.getElementById("mac").value;

const regex = /^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$/;

if (regex.test(mac)) {

print("✅ IoT Device Authenticated.");

ledger.push("IoT MAC verified: " + mac);

} else {

print("❌ Invalid MAC Address.");

ledger.push("IoT Auth failed: " + mac);

}

}

function verifyBlockchain() {

const cert = document.getElementById("bcert").value;

if (cert.startsWith("BC") && cert.length === 8) {

print("✅ Blockchain Certificate Verified.");

ledger.push("Blockchain cert: " + cert);

} else {

print("❌ Blockchain Certificate Invalid.");

ledger.push("Blockchain cert failed: " + cert);

}

}

</script>

</body>

</html>