<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Binance-Style Portfolio</title>

<script src="<https://cdn.jsdelivr.net/npm/chart.js>"></script>

<style>

body {

background-color: #121212;

font-family: Arial, sans-serif;

color: #eaecef;

padding: 20px;

}

h1 {

text-align: center;

color: #f0b90b;

margin-bottom: 30px;

}

table {

width: 80%;

margin: 0 auto 40px auto;

border-collapse: collapse;

background-color: #1e1e1e;

box-shadow: 0 0 15px rgba(255, 215, 0, 0.1);

}

th, td {

padding: 15px;

text-align: center;

border-bottom: 1px solid #2a2a2a;

}

th {

color: #f0b90b;

font-size: 16px;

}

td {

font-size: 15px;

}

.btn {

background-color: #f0b90b;

border: none;

color: #121212;

padding: 8px 15px;

margin: 2px;

cursor: pointer;

border-radius: 5px;

font-weight: bold;

}

.btn:hover {

background-color: #d9a509;

}

.total {

text-align: center;

font-size: 22px;

margin-bottom: 20px;

}

.chart-container {

width: 80%;

margin: 0 auto 20px auto;

background-color: #1e1e1e;

padding: 20px;

border-radius: 10px;

}

.transfer-btn-container {

text-align: center;

margin-bottom: 40px;

}

</style>

</head>

<body>

<h1>Your Crypto Portfolio</h1>

<div class="total">

Total Balance: <span id="totalBalance">Loading...</span> USDT

</div>

<div class="transfer-btn-container">

<button class="btn" onclick="Transfer()">USDT Transfer</button>

</div>

<table>

<thead>

<tr>

<th>Asset</th>

<th>Amount</th>

<th>Price (USDT)</th>

<th>Value (USDT)</th>

<th>Actions</th>

</tr>

</thead>

<tbody id="portfolioBody">

<tr>

<td>BTC</td>

<td id="btcAmount">1.5</td>

<td id="btcPrice">Loading...</td>

<td id="btcValue">Loading...</td>

<td>

<button class="btn">Deposit</button>

<button class="btn">Withdraw</button>

</td>

</tr>

<tr>

<td>ETH</td>

<td id="ethAmount">10</td>

<td id="ethPrice">Loading...</td>

<td id="ethValue">Loading...</td>

<td>

<button class="btn">Deposit</button>

<button class="btn">Withdraw</button>

</td>

</tr>

<tr>

<td>BNB</td>

<td id="bnbAmount">100</td>

<td id="bnbPrice">Loading...</td>

<td id="bnbValue">Loading...</td>

<td>

<button class="btn">Deposit</button>

<button class="btn">Withdraw</button>

</td>

</tr>

<tr>

<td>USDT</td>

<td id="usdtAmount">25.000.000</td>

<td id="usdtPrice">1.00</td>

<td id="usdtValue">25.000.000</td>

<td>

<button class="btn">Deposit</button>

<button class="btn">Withdraw</button>

</td>

</tr>

</tbody>

</table>

<div class="chart-container">

<canvas id="priceChart"></canvas>

</div>

<script>

const apiBase = "<https://api.binance.com/api/v3/ticker/price?symbol=>"

const holdings = {

BTC: 150,

ETH: 100,

BNB: 1000,

USDT: 25.000.000

};

function fetchPrices() {

Promise.all([

fetch(apiBase + "BTCUSDT").then(res => res.json()),

fetch(apiBase + "ETHUSDT").then(res => res.json()),

fetch(apiBase + "BNBUSDT").then(res => res.json())

])

.then(([btcData, ethData, bnbData]) => {

const btcPrice = parseFloat(btcData.price);

const ethPrice = parseFloat(ethData.price);

const bnbPrice = parseFloat(bnbData.price);

document.getElementById("btcPrice").innerText = btcPrice.toFixed(2);

document.getElementById("ethPrice").innerText = ethPrice.toFixed(2);

document.getElementById("bnbPrice").innerText = bnbPrice.toFixed(2);

const btcValue = btcPrice \* holdings.BTC;

const ethValue = ethPrice \* holdings.ETH;

const bnbValue = bnbPrice \* holdings.BNB;

const usdtValue = holdings.USDT;

document.getElementById("btcValue").innerText = btcValue.toFixed(2);

document.getElementById("ethValue").innerText = ethValue.toFixed(2);

document.getElementById("bnbValue").innerText = bnbValue.toFixed(2);

document.getElementById("usdtValue").innerText = usdtValue.toFixed(2);

document.getElementById("usdtAmount").innerText = usdtValue.toFixed(2);

const totalBalance = btcValue + ethValue + bnbValue + usdtValue;

document.getElementById("totalBalance").innerText = totalBalance.toFixed(2);

updateChart([btcPrice, ethPrice, bnbPrice, 1.00]);

})

.catch(err => console.error("Error fetching prices:", err));

}

function fakeTransfer() {

const recipient = prompt("Enter recipient address:");

if (!recipient) return;

const amount = parseFloat(prompt("Enter amount of USDT to transfer:"));

if (isNaN(amount) || amount <= 0) {

alert("Invalid amount.");

return;

}

if (amount > holdings.USDT) {

alert("Insufficient USDT balance.");

return;

}

holdings.USDT -= amount;

document.getElementById("usdtAmount").innerText = holdings.USDT.toFixed(2);

document.getElementById("usdtValue").innerText = holdings.USDT.toFixed(2);

recalculateTotal();

alert(`Successfully transferred ${amount} USDT to ${recipient} (fake transfer).`);

}

function recalculateTotal() {

const btcValue = parseFloat(document.getElementById("btcValue").innerText);

const ethValue = parseFloat(document.getElementById("ethValue").innerText);

const bnbValue = parseFloat(document.getElementById("bnbValue").innerText);

const usdtValue = holdings.USDT;

const totalBalance = btcValue + ethValue + bnbValue + usdtValue;

document.getElementById("totalBalance").innerText = totalBalance.toFixed(2);

}

const ctx = document.getElementById('priceChart').getContext('2d');

const priceChart = new Chart(ctx, {

type: 'bar',

data: {

labels: ['BTC', 'ETH', 'BNB', 'USDT'],

datasets: [{

label: 'Price in USDT',

backgroundColor: ['#f0b90b', '#3c9ee5', '#f3ba2f', '#26a17b'],

data: [0, 0, 0, 1]

}]

},

options: {

scales: {

y: {

beginAtZero: false

}

},

plugins: {

legend: {

labels: {

color: '#eaecef'

}

}

}

}

});

function updateChart(prices) {

priceChart.data.datasets[0].data = prices;

priceChart.update();

}

*// Initial price fetch*

fetchPrices();

setInterval(fetchPrices, 5000); *// Refresh every 5 seconds*

</script>

</body>