Assignment 7

Q.Write a program to design Cricket Scorecard website (Live score display) using XML AND JSON

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Live Cricket Score</title>

<style>

body {

font-family: 'Arial', sans-serif;

background-color: #008080;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

color: #333;

text-align: center;

}

.scoreboard {

background-color: #000000;

border-radius: 15px;

padding: 30px;

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

width: 400px;

transition: transform 0.3s ease-in-out;

}

.scoreboard:hover {

transform: scale(1.05);

}

h1 {

font-size: 40px;

color: #0277bd;

font-weight: bold;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);

margin-bottom: 20px;

}

.team {

margin: 20px 0;

padding: 10px;

background-color: #ffffff;

border-radius: 10px;

box-shadow: 0 0 8px rgba(0, 0, 0, 0.1);

transition: background-color 0.3s ease;

}

.team:hover {

background-color: #80deea;

}

h2 {

font-size: 28px;

color: #FF5733 ;

margin-bottom: 10px;

}

h3 {

font-size: 28px;

color: #FF0000 ;

margin-bottom: 10px;

}

.score {

font-size: 30px;

font-weight: bold;

margin: 10px 0;

color: #800000;

text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.2);

}

.players {

font-size: 18px;

color: #455a64;

margin-top: 15px;

}

.player {

margin: 8px 0;

padding: 8px;

background-color: #f1f8e9;

border-radius: 8px;

transition: background-color 0.3s ease;

}

.player:hover {

background-color: #c8e6c9;

cursor: pointer;

}

.refresh-btn {

margin-top: 20px;

padding: 12px 20px;

background-color: #008000;

color: white;

border: none;

border-radius: 5px;

font-size: 18px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.refresh-btn:hover {

background-color: #0277bd;

}

</style>

</head>

<body>

<div class="scoreboard">

<h1>Live Cricket Score</h1>

<div id="teamA" class="team">

<h2>INDIA </h2>

<div class="score"></div>

<div class="players"></div>

</div>

<div id="teamB" class="team">

<h3>ENGLAND</h3>

<div class="score"></div>

<div class="players"></div>

</div>

<button class="refresh-btn" onclick="refreshScore()">Refresh Score</button>

</div>

<script>

const xmlData = `

<match>

<teamA>

<name>Team A</name>

<score>250/6</score>

<players>

<player>

<name>Player 1</name>

<runs>40</runs>

<balls>30</balls>

</player>

<player>

<name>Player 2</name>

<runs>50</runs>

<balls>40</balls>

</player>

</players>

</teamA>

<teamB>

<name>Team B</name>

<score>180/5</score>

<players>

<player>

<name>Player 1</name>

<runs>60</runs>

<balls>50</balls>

</player>

<player>

<name>Player 2</name>

<runs>25</runs>

<balls>20</balls>

</player>

</players>

</teamB>

</match>

`;

function parseXMLAndUpdateScoreboard(xmlString) {

const parser = new DOMParser();

const xmlDoc = parser.parseFromString(xmlString, "text/xml");

const teamA = xmlDoc.getElementsByTagName("teamA")[0];

const teamAScore = teamA.getElementsByTagName("score")[0].textContent;

const teamAPlayers = teamA.getElementsByTagName("player");

const teamADiv = document.getElementById("teamA");

teamADiv.querySelector(".score").textContent = `Score: ${teamAScore}`;

const teamAPlayersDiv = teamADiv.querySelector(".players");

teamAPlayersDiv.innerHTML = ""; // Clear existing players

for (let player of teamAPlayers) {

const playerName = player.getElementsByTagName("name")[0].textContent;

const playerRuns = player.getElementsByTagName("runs")[0].textContent;

const playerBalls = player.getElementsByTagName("balls")[0].textContent;

const playerDiv = document.createElement("div");

playerDiv.classList.add("player");

playerDiv.textContent = `${playerName} - ${playerRuns} runs (${playerBalls} balls)`;

teamAPlayersDiv.appendChild(playerDiv);

}

const teamB = xmlDoc.getElementsByTagName("teamB")[0];

const teamBScore = teamB.getElementsByTagName("score")[0].textContent;

const teamBPlayers = teamB.getElementsByTagName("player");

const teamBDiv = document.getElementById("teamB");

teamBDiv.querySelector(".score").textContent = `Score: ${teamBScore}`;

const teamBPlayersDiv = teamBDiv.querySelector(".players");

teamBPlayersDiv.innerHTML = ""; // Clear existing players

for (let player of teamBPlayers) {

const playerName = player.getElementsByTagName("name")[0].textContent;

const playerRuns = player.getElementsByTagName("runs")[0].textContent;

const playerBalls = player.getElementsByTagName("balls")[0].textContent;

const playerDiv = document.createElement("div");

playerDiv.classList.add("player");

playerDiv.textContent = `${playerName} - ${playerRuns} runs (${playerBalls} balls)`;

teamBPlayersDiv.appendChild(playerDiv);

}

}

parseXMLAndUpdateScoreboard(xmlData);

function fetchNewData() {

const newXMLData = `

<match>

<teamA>

<name>Team A</name>

<score>${Math.floor(Math.random() \* 300)}/${Math.floor(Math.random() \* 10)}</score>

<players>

<player>

<name>Player 1</name>

<runs>${Math.floor(Math.random() \* 100)}</runs>

<balls>${Math.floor(Math.random() \* 60)}</balls>

</player>

<player>

<name>Player 2</name>

<runs>${Math.floor(Math.random() \* 100)}</runs>

<balls>${Math.floor(Math.random() \* 60)}</balls>

</player>

</players>

</teamA>

<teamB>

<name>Team B</name>

<score>${Math.floor(Math.random() \* 300)}/${Math.floor(Math.random() \* 10)}</score>

<players>

<player>

<name>Player 1</name>

<runs>${Math.floor(Math.random() \* 100)}</runs>

<balls>${Math.floor(Math.random() \* 60)}</balls>

</player>

<player>

<name>Player 2</name>

<runs>${Math.floor(Math.random() \* 100)}</runs>

<balls>${Math.floor(Math.random() \* 60)}</balls>

</player>

</players>

</teamB>

</match>

`;

return newXMLData;

}

function refreshScore() {

const newXMLData = fetchNewData(); // Simulate fetching new data

parseXMLAndUpdateScoreboard(newXMLData); // Update the scoreboard with new data

}

</script>

</body>