Internet Programming (CSYM019-SPR) - TASK 1 explanation

Poovarasan Sandhanam - 23856814

Overview:

This application is a Football Premier League-based web application. It consists of three main components: the fixtures table, team rankings table, and top scorers table. Users can navigate between these components using buttons provided on each page. The data for fixtures, top scorers, and team rankings is fetched from a JSON file and dynamically rendered onto the web page using JavaScript. The styling of the application is done using CSS.

Introduction:

The HTML files (league.html and topscorers.html) provide the structure of the web pages, including the layout, headings, and placeholders for dynamic content. The CSS file (league.css) styles the elements of the web pages, ensuring a visually appealing and consistent design. The JavaScript file (league.js) fetches data from a JSON file(league.json), processes it, and dynamically generates tables for fixtures, team rankings, and top scorers.

Explanation:

league.html

<!DOCTYPE html> <!-- Declaring the document type as HTML5 -->

<html lang="en"><!--The <html> tag sets the language of the document to English.-->

<head>

<meta charset="UTF-8">

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

<title>Team Rankings</title> <!-- Setting the title of the web page -->

<link rel="stylesheet" href="league.css"> <!-- Linking to external CSS file -->

</head>

<body>

<div class="team-table-container"> <!-- Container for fixtures table -->

<h2>Fixtures Table</h2> <!-- Heading for fixtures table -->

<div id="fixtures-table"></div> <!-- Placeholder for fixtures table -->

</div>

<div class="team-table-container"> <!-- Container for team rankings table -->

<h2>Team Rankings</h2> <!-- Heading for team rankings table -->

<div id="team-rankings"></div> <!-- Placeholder for team rankings table -->

</div>

<!-- Button to navigate to topscores.html -->

<div>

<button id="button-container">Go to Top Scorers</button> <!-- Button for navigating to top scorers page -->

</div>

<script src="league.js"></script> <!-- Linking to external JavaScript file -->

<script>

// Add event listener for the button

document.getElementById('button-container').addEventListener('click', function() {

window.location.href = 'topscorers.html'; // Redirecting to topscorers.html when button is clicked

});

</script>

</body>

</html>

topscorer.html

<!DOCTYPE html> <!-- Declaring the document type as HTML5 -->

<html lang="en">

<head>

<meta charset="UTF-8"> <!-- Declaring character encoding as UTF-8 -->

<meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Setting viewport for responsive design -->

<title>Top Scorers</title> <!-- Setting the title of the web page -->

<link rel="stylesheet" href="league.css"> <!-- Linking to external CSS file for styling -->

</head>

<body>

<h1>Top Scorers of the League</h1> <!-- Heading for the top scorers section -->

<div id="top-scorers"></div> <!-- Placeholder for displaying top scorers -->

<script src="league.js"></script> <!-- Linking to external JavaScript file for dynamic functionality -->

</body>

</html>

League.css

/\* Reset default margins and paddings \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body styling \*/

body {

font-family: Arial, sans-serif; /\* Set font family \*/

background-color:#bcbb93; /\* Set background color \*/

}

/\* Container for button \*/

#button-container {

position: fixed; /\* Position the button \*/

top: 20px; /\* Distance from the top \*/

right: 20px; /\* Distance from the right \*/

z-index: 999; /\* Adjust the stacking order \*/

background-color: #032447; /\* Background color \*/

color: white; /\* Text color \*/

padding: 10px 20px; /\* Padding \*/

border-radius: 5px; /\* Rounded corners \*/

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2); /\* Shadow effect \*/

}

/\* Heading styles \*/

h1 {

text-align: center; /\* Center align heading \*/

margin-top: 30px; /\* Margin from the top \*/

color: #150959; /\* Text color \*/

}

h2 {

text-align: center; /\* Center align heading \*/

color: #050561; /\* Text color \*/

}

/\* Table styles \*/

table {

width: 100%; /\* Full width \*/

border-collapse: collapse; /\* Collapse border spacing \*/

margin: 20px auto; /\* Center the table horizontally \*/

background-color: #fff; /\* White background \*/

border-radius: 8px; /\* Rounded corners \*/

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Shadow effect \*/

border: 1px solid #ddd; /\* Border \*/

}

th,

td {

padding: 12px; /\* Padding \*/

text-align: left; /\* Left align text \*/

border-bottom: 1px solid #ddd; /\* Bottom border \*/

border-right: 1px solid #ddd; /\* Right border \*/

}

caption {

font-size: 1.5em; /\* Font size \*/

margin-bottom: 10px; /\* Margin at the bottom \*/

text-align: center; /\* Center align text \*/

}

/\* Top scorers table \*/

#top-scorers td:first-child {

padding-left: 20px; /\* Left padding \*/

}

/\* Hover effect for table rows \*/

tr:hover {

background-color: #f2f2f2; /\* Light gray background on hover \*/

}

/\* Fixture list \*/

.fixture-list {

display: flex; /\* Flex container \*/

flex-wrap: wrap; /\* Wrap items \*/

gap: 20px; /\* Gap between items \*/

justify-content: center; /\* Center align items \*/

margin: 20px auto; /\* Center the fixture list horizontally \*/

}

.fixture {

background-color: #fff; /\* White background \*/

border-radius: 8px; /\* Rounded corners \*/

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Shadow effect \*/

padding: 15px; /\* Padding \*/

text-align: center; /\* Center align text \*/

width: 200px; /\* Width of fixture \*/

transition: transform 0.3s ease; /\* Smooth transition on hover \*/

}

.fixture:hover {

transform: translateY(-5px); /\* Move up slightly on hover \*/

}

/\* Team table container \*/

.team-table-container {

justify-content: center; /\* Center align content \*/

margin: 100px; /\* Margin \*/

}

/\* Additional styling for top scorers table \*/

#top-scorers {

margin: 20px auto; /\* Margin \*/

max-width: 800px; /\* Maximum width \*/

}

/\* Styling for top scorers table \*/

#top-scorers table {

width: 100%; /\* Full width \*/

border-collapse: collapse; /\* Collapse border spacing \*/

background-color: #eaf6ff; /\* Light blue background \*/

border-radius: 8px; /\* Rounded corners \*/

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Shadow effect \*/

border: 1px solid #ddd; /\* Border \*/

}

#top-scorers th,

#top-scorers td {

padding: 12px; /\* Padding \*/

text-align: left; /\* Left align text \*/

border-bottom: 1px solid #ddd; /\* Bottom border \*/

border-right: 1px solid #ddd; /\* Right border \*/

}

#top-scorers caption {

font-size: 1.5em; /\* Font size \*/

margin-bottom: 10px; /\* Margin at the bottom \*/

text-align: center; /\* Center align text \*/

color: #007bff; /\* Blue heading color \*/

}

#top-scorers tr:hover {

background-color: #d4eaff; /\* Light blue background on hover \*/

}

#top-scorers td:first-child {

padding-left: 20px; /\* Left padding \*/

}

/\* Additional styling for top teams table \*/

#top-teams {

margin: 20px auto; /\* Margin \*/

max-width: 800px; /\* Maximum width \*/

}

/\* Styling for top teams table \*/

#top-teams table {

width: 100%; /\* Full width \*/

border-collapse: collapse; /\* Collapse border spacing \*/

background-color: #ffe6e6; /\* Light red background \*/

border-radius: 8px; /\* Rounded corners \*/

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /\* Shadow effect \*/

border: 1px solid #ddd; /\* Border \*/

}

#top-teams th,

#top-teams td {

padding: 12px; /\* Padding \*/

text-align: left; /\* Left align text \*/

border-bottom: 1px solid #ddd; /\* Bottom border \*/

border-right: 1px solid #ddd; /\* Right border \*/

}

#top-teams caption {

font-size: 1.5em; /\* Font size \*/

margin-bottom: 10px; /\* Margin at the bottom \*/

text-align: center; /\* Center align text \*/

color: #ff0000; /\* Red heading color \*/

}

#top-teams tr:hover {

background-color: #ffd9d9; /\* Light red background on hover \*/

}

#top-teams td:first-child {

padding-left: 20px; /\* Left padding \*/

}

League.js

// Event listener for when the HTML document is fully loaded

document.addEventListener("DOMContentLoaded", function() {

// Fetch league data from league.json using Fetch API

fetch("league.json")

// Convert response to JSON format

.then(response => response.json())

// Handle the fetched data

.then(data => {

// Extract fixtures, team data, and top scorers data from fetched JSON

const fixtures = data.fixtures;

const teamData = data.teams;

const scorerData = data.topScorers;

// Sort teams by points in descending order

teamData.sort((a, b) => b.points - a.points);

// Assign positions dynamically based on sorted data

assignPositions(teamData);

// Sort top scorers by goals in descending order

scorerData.sort((a, b) => b.goals - a.goals);

// Assign positions to top scorers based on sorted data

assignPositions(scorerData);

// Generate tables for team rankings and top scorers

const teamTable = tableContainer(teamData, "", true);

const scorerTable = tableContainer(scorerData, "", false);

// Get placeholders for team rankings, top scorers, and fixtures

const team = document.querySelector("#team-rankings");

const scorer = document.querySelector("#top-scorers");

const fixture = document.querySelector("#fixtures-table");

// Append team rankings table to its placeholder

if (team) {

team.appendChild(teamTable);

} else {

console.error('Team rankings placeholder not found.');

}

// Append top scorers table to its placeholder

if (scorer) {

scorer.appendChild(scorerTable);

} else {

console.error('Top scorers placeholder not found.');

}

// Generate fixture table and append it to its placeholder

if (fixture) {

const fixtureList = fixtureListContainer(fixtures);

fixture.appendChild(fixtureList);

} else {

console.error('Fixtures table placeholder not found.');

}

})

// Handle errors during data fetching

.catch(error => {

console.error('Error fetching data:', error);

});

});

// Function to generate table for fixture data

function fixtureListContainer(fixtures) {

const table = document.createElement("table"); // Create table element

table.className = "fixture-table"; // Add class name for styling

const headerRow = table.insertRow(); // Insert row for table header

const headers = ["Home Team", "Score", "Away Team"]; // Header text array

// Create header cells and append to header row

headers.forEach(headerText => {

const th = document.createElement("th");

th.textContent = headerText;

headerRow.appendChild(th);

});

// Iterate through fixtures and populate table rows

fixtures.forEach(fixture => {

const row = table.insertRow(); // Insert new row

// Insert cells for home team, score, and away team

const homeTeamCell = row.insertCell();

homeTeamCell.textContent = fixture.homeTeam;

const scoreCell = row.insertCell();

scoreCell.textContent = ${fixture.homeScore} - ${fixture.awayScore};

const awayTeamCell = row.insertCell();

awayTeamCell.textContent = fixture.awayTeam;

});

return table; // Return generated table

}

// Function to generate dynamic table based on data

function tableContainer(data, title, includeExtraColumns) {

const table = document.createElement("table"); // Create table element

table.style.width = "100%"; // Set table width to 100%

table.style.textAlign = "center"; // Center align content

table.style.borderCollapse = "collapse"; // Collapse table borders

const caption = document.createElement("caption"); // Create table caption

caption.textContent = title; // Set caption text

table.appendChild(caption); // Append caption to table

const headerRow = table.insertRow(); // Insert row for table header

const headers = includeExtraColumns

? ["Position", "Team", "Played Games", "Won", "Drawn", "Lost", "Goal Difference", "Points", "Form"]

: ["Position","Player (Team)", "Goals", "Assists", "Played", "Goals per 90", "Mins per Goal", "Total Shots", "Goal Conversion", "Shot Accuracy"];

// Create header cells and append to header row

headers.forEach(headerText => {

const th = document.createElement("th");

th.textContent = headerText;

th.style.border = "1px solid #ddd"; // Add border for table header

th.style.padding = "8px"; // Add padding for table header

headerRow.appendChild(th);

});

// Iterate through data and populate table rows

data.forEach(item => {

const row = table.insertRow(); // Insert new row

row.style.border = "1px solid #ddd"; // Add border for table row

let rowData;

if (item.name) {

// Create a cell for the team logo and name

const teamCell = row.insertCell();

teamCell.style.display = "flex";

teamCell.style.alignItems = "center";

teamCell.style.justifyContent = "center"; // Center content

// Create an image element for the logo

const logo = document.createElement("img");

logo.src = item.image;

logo.alt = item.name;

logo.style.maxWidth = "30px"; // Set maximum width for logo

logo.style.marginRight = "8px"; // Add margin between image and text

// Append the logo and team name to the cell

teamCell.appendChild(logo);

teamCell.appendChild(document.createTextNode(item.name));

const formIcons = createFormIcons(item.lastSixGames); // Create form icons based on last six games

rowData = [item.position, teamCell, item.playedGames, item.won, item.drawn, item.lost, item.goalDifference, item.points, formIcons];

} else {

rowData = [item.position,${item.player} (${item.team}), item.goals, item.assists, item.played, item.goalsPer90, item.minsPerGoal, item.totalShots, item.goalConversion, item.shotAccuracy];

}

// Create cells for table data and append to row

rowData.forEach(dataItem => {

const cell = row.insertCell();

if (dataItem instanceof Node) {

cell.appendChild(dataItem);

} else {

cell.textContent = dataItem;

}

cell.style.padding = "8px"; // Add padding for all cells

});

});

return table; // Return generated table

}

// Function to create form icons based on last six games

function createFormIcons(lastSixGames) {

const container = document.createElement("div"); // Create container for icons

container.style.display = "flex"; // Set display property to flex

container.style.justifyContent = "center"; // Center align icons

// Iterate through last six games data and create icons

lastSixGames.split('').forEach(game => {

const icon = document.createElement("span"); // Create span element for icon

icon.style.display = “inline-block";

League.json

{

"fixtures": [

{

"homeTeam": "Arsenal", // Home team for the match

"awayTeam": "Manchester City", // Away team for the match

"homeScore": 2, // Score of the home team

"awayScore": 1 // Score of the away team

},

// More fixtures follow with similar structure

],

"topScorers": [

{

"position": 1, // Position in the top scorers list

"player": "Cole Palmer", // Player's name

"team": "Chelsea, Manchester City", // Team(s) the player belongs to

"goals": 22, // Number of goals scored by the player

"assists": 10, // Number of assists by the player

"played": 33, // Number of games played by the player

"goalsPer90": 0.78, // Goals scored per 90 minutes played

"minsPerGoal": 115, // Average minutes taken to score a goal

"totalShots": 75, // Total number of shots taken by the player

"goalConversion": "29%", // Conversion rate of shots into goals

"shotAccuracy": "61%" // Accuracy of shots on target

},

// More top scorers follow with similar structure

],

"teams": [

{

"position": 1, // Position in the league table

"name": "Arsenal", // Team name

"image":"./asset/arsenal.png", // URL of the team's logo image

"playedGames": 28, // Number of games played by the team

"won": 20, // Number of games won by the team

"drawn": 4, // Number of games drawn by the team

"lost": 4, // Number of games lost by the team

"for": 58, // Goals scored by the team

"against": 22, // Goals conceded by the team

"goalDifference": 36, // Difference between goals scored and conceded

"points": 64, // Total points earned by the team

"lastSixGames": "WWLLWL" // Form of the team in the last six games (W - Win, L - Loss, D - Draw)

},

// More teams follow with similar structure

]

}