**CSYM019: Internet Programming  
Assignment**   
  
Name : Rathnayake Mudiyanselage Sanindu Madhushan Rathnayake

Student ID: 23849711

Code Explanation

Part - 01

**Task 01**

**JSON schema  
Premier League Table:**

This section represents the league standings, typically organized as an array of objects.

Each object contains information about a team's performance, including the team's name, number of games played, games won, games drawn, games lost, goals scored (for), goals conceded (against), goal difference (gd), and total points earned.

All these properties are required.

**Top Scorers:**

This section lists the top goal scorers in the league.

It is also an array of objects.

Each object includes details such as the player's name, their team, total goals scored, total assists, number of games played, goals per 90 minutes, minutes per goal, total shots taken, goal conversion rate, and shot accuracy.

Similar to the Premier League Table, all properties are required.

**Scores Fixtures:**

This part deals with match fixtures and results.

It's structured as an array of objects, with each object representing a date and the matches played on that date.

Each match object contains details about the teams involved, their scores, and the conclusion status (typically "FT" for full-time).

Required properties include the date of the fixtures and the matches played.

Overall, this schema provides a structured way to represent key information about the Premier League, including standings, player statistics, and match results.

**League.js**

League.js  
  
// Function to make Ajax request

function makeAjaxRequest(url, callback) {

    var xhr = new XMLHttpRequest();

    xhr.onreadystatechange = function() {

        // Check if the request is complete

        if (xhr.readyState === 4) {

            // Check if the request was successful

            if (xhr.status === 200) {

                try {

                    // Parse the JSON response

                    var data = JSON.parse(xhr.responseText);

                    // Call the callback function with the parsed data

                    callback(null, data);

                } catch (error) {

                    // Call the callback function with the error

                    callback(error, null);

                }

            } else {

                // Call the callback function with an error if the request failed

                callback(new Error('Request failed'), null);

            }

        }

    };

    // Open a GET request to the specified URL

    xhr.open('GET', url, true);

    // Send the request

    xhr.send();

}

// Calculate points for each team

function calculatePoints(won, drawn) {

    return (won \* 3) + drawn;

}

// Update the points in the league table data

function updatePoints(data) {

    // Iterate through each team and update their points

    data.forEach(function(team) {

        team.points = calculatePoints(team.won, team.drawn);

    });

}

// Fetch Premier League table data from JSON file

function fetchPremierLeagueTableData() {

    // Make an Ajax request to fetch the league table data

    makeAjaxRequest('League.json', function(error, data) {

        if (error) {

            // Log an error message if data fetching fails

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

        } else {

            // Update the points for each team

            updatePoints(data.premier\_league\_table);

            // Display the updated league table

            displayPremierLeagueTable(data.premier\_league\_table);

        }

    });

}

// Display Premier League table data

function displayPremierLeagueTable(data) {

    // Get the table element by its ID

    var table = document.getElementById('premier-league-table');

    // Set the table headers

    table.innerHTML = '<thead><tr><th>Team</th><th>Played</th><th>Won</th><th>Drawn</th><th>Lost</th><th>For</th><th>Against</th><th>GD</th><th>Points</th></tr></thead>';

    // Create a new tbody element

    var tbody = document.createElement('tbody');

    // Loop through the data array and create table rows for each team

    data.forEach(function(team) {

        var row = `<tr>

                        <td>${team.team}</td>

                        <td>${team.played}</td>

                        <td>${team.won}</td>

                        <td>${team.drawn}</td>

                        <td>${team.lost}</td>

                        <td>${team.for}</td>

                        <td>${team.against}</td>

                        <td>${team.gd}</td>

                        <td>${team.points}</td>

                    </tr>`;

        // Add the row to the tbody

        tbody.innerHTML += row;

    });

    // Append the tbody to the table

    table.appendChild(tbody);

}

// Update Premier League table every 1 hour

function updateTablePeriodically() {

    // Fetch and display the Premier League table data

    fetchPremierLeagueTableData();

    // Set a timeout to update the table again after  1 hour

    setTimeout(updateTablePeriodically, 3600000);

}

// Initial setup for Premier League table

function initPremierLeagueTable() {

    // Fetch and display the Premier League table data

    fetchPremierLeagueTableData();

    // Update the Premier League table periodically

    updateTablePeriodically();

}

// Call the init function when the window loads

window.onload = function() {

    try {

        // Initialize the Premier League table setup

        initPremierLeagueTable();

    } catch (error) {

        // Log any errors that occur during initialization

        console.error('An error occurred:', error);

    }

};

**League.css**

League.css  
  
/\* General styles for the entire webpage \*/

body {

    font-family: Arial, sans-serif; /\* Sets the font for the webpage \*/

    margin: 0; /\* Removes the default margin \*/

    padding: 0; /\* Removes the default padding \*/

}

/\* Styles for the header section \*/

header {

    background-color: #333; /\* Dark background color for the header \*/

    color: #fff; /\* White text color for the header \*/

    padding: 20px; /\* Adds padding inside the header \*/

    text-align: center; /\* Centers the header text \*/

}

/\* Styles for the main content area \*/

main {

    padding: 20px; /\* Adds padding inside the main content area \*/

}

/\* Styles for the footer section \*/

footer {

    background-color: #333; /\* Dark background color for the footer \*/

    color: #fff; /\* White text color for the footer \*/

    padding: 10px 20px; /\* Adds padding inside the footer \*/

    text-align: center; /\* Centers the footer text \*/

    position: fixed; /\* Fixes the footer at the bottom of the page \*/

    bottom: 0; /\* Positions the footer at the very bottom \*/

    width: 100%; /\* Makes the footer span the entire width of the page \*/

}

/\* Style for the Premier League table \*/

#premier-league-table {

    width: 100%; /\* Makes the table span the entire width of its container \*/

    border-collapse: collapse; /\* Collapses table borders into a single border \*/

}

/\* Styles for table headers and cells \*/

#premier-league-table th,

#premier-league-table td {

    border: 1px solid #ddd; /\* Adds a light gray border around cells \*/

    padding: 8px; /\* Adds padding inside table cells \*/

    text-align: center; /\* Centers the text inside table cells \*/

}

/\* Styles for table headers \*/

#premier-league-table th {

    background-color: #f2f2f2; /\* Light gray background color for headers \*/

    color: #333; /\* Dark gray text color for headers \*/

}

/\* Styles for even-numbered table rows \*/

#premier-league-table tr:nth-child(even) {

    background-color: #f2f2f2; /\* Light gray background color for even rows \*/

}

/\* Styles for table rows on hover \*/

#premier-league-table tr:hover {

    background-color: #ddd; /\* Darker gray background color when hovering over a row \*/

}

**League.html**

**League.html**<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <!-- Title of the webpage -->

    <title>Premier League Dashboard</title>

    <!-- Link to the external CSS file for styling -->

    <link rel="stylesheet" href="League.css">

</head>

<body>

    <!-- Header section containing the main title -->

    <header>

        <h1>Premier League Dashboard</h1>

    </header>

    <!-- Main content section -->

    <main>

        <!-- Section for displaying the Premier League table -->

        <section id="premier-league-section">

            <!-- Table to display Premier League data, content will be dynamically generated -->

            <table id="premier-league-table">

                <!-- Table content will be dynamically generated here -->

            </table>

        </section>

    </main>

    <!-- Adding some space before the footer -->

    <br><br><br><br>

    <!-- Footer section with a copyright notice -->

    <footer>

        <p>&copy; 2024 Premier League Dashboard</p>

    </footer>

    <!-- Link to the external JavaScript file for functionality -->

    <script src="League.js"></script>

</body>

</html>

**Topscore.js**

// Function to make an Ajax request to the specified URL and execute a callback function with the retrieved data

function makeAjaxRequest(url, callback) {

    // Create a new XMLHttpRequest object

    var xhr = new XMLHttpRequest();

    // Define a function to handle state changes of the request

    xhr.onreadystatechange = function() {

        // When the request is completed

        if (xhr.readyState === 4) {

            // If the request was successful

            if (xhr.status === 200) {

                try {

                    // Parse the response JSON data

                    var data = JSON.parse(xhr.responseText);

                    // Call the callback function with the retrieved data

                    callback(null, data);

                } catch (error) {

                    // If parsing the response JSON data fails, call the callback function with the error

                    callback(error, null);

                }

            } else {

                // If the request fails, call the callback function with an error

                callback(new Error('Request failed'), null);

            }

        }

    };

    // Open the request with the specified method and URL

    xhr.open('GET', url, true);

    // Send the request

    xhr.send();

}

// Fetch top scorers data from a JSON file and update the table periodically

function fetchTopScorersData() {

    // Make an Ajax request to retrieve data from the 'League.json' file

    makeAjaxRequest('League.json', function(error, data) {

        if (error) {

            // If an error occurs during the request, log the error

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

        } else {

            // If data is successfully retrieved, display the top scorers

            displayTopScorers(data.top\_scorers);

            // Set a timeout to fetch the data again after 1 hour (3600000 milliseconds)

            setTimeout(fetchTopScorersData, 3600000);

        }

    });

}

// Display top scorers data in a table

function displayTopScorers(data) {

    // Get the table element

    var table = document.getElementById('top-scorers-table');

    // Create table header with column names

    table.innerHTML = '<thead><tr><th>Name</th><th>Team</th><th>Goals</th><th>Assists</th><th>Played</th><th>Goals/90</th><th>Mins/Goal</th><th>Total Shots</th><th>Goal Conversion</th><th>Shot Accuracy</th></tr></thead>';

    // Create a new tbody element

    var tbody = document.createElement('tbody');

    // Iterate over each player data and create table rows

    data.forEach(function(player) {

        var row = `<tr>

                        <td>${player.name}</td>

                        <td>${player.team}</td>

                        <td>${player.goals}</td>

                        <td>${player.assists}</td>

                        <td>${player.played}</td>

                        <td>${player.goals\_per\_90}</td>

                        <td>${player.mins\_per\_goal}</td>

                        <td>${player.total\_shots}</td>

                        <td>${player.goal\_conversion}</td>

                        <td>${player.shot\_accuracy}</td>

                    </tr>`;

        // Append each row to the tbody

        tbody.innerHTML += row;

    });

    // Append the tbody to the table

    table.appendChild(tbody);

}

// Initialize the top scorers functionality

function initTopScorers() {

    // Start fetching top scorers data

    fetchTopScorersData();

}

// Call the init function when the window loads

window.onload = function() {

    try {

        // Initialize the top scorers functionality

        initTopScorers();

    } catch (error) {

        // If an error occurs during initialization, log the error

        console.error('An error occurred:', error);

    }

};

**Topscore.css**

**Topscore.css**/\* General styles for the entire webpage \*/

body {

    font-family: Arial, sans-serif; /\* Sets the font for the webpage \*/

    margin: 0; /\* Removes the default margin \*/

    padding: 0; /\* Removes the default padding \*/

    background-color: #f9f9f9; /\* Sets a light gray background color \*/

    color: #333; /\* Sets the default text color to a dark gray \*/

}

/\* Styles for the header section \*/

header {

    background-color: #333; /\* Dark background color for the header \*/

    color: #fff; /\* White text color for the header \*/

    padding: 20px; /\* Adds padding inside the header \*/

    text-align: center; /\* Centers the header text \*/

}

/\* Styles for the main content area \*/

main {

    padding: 20px; /\* Adds padding inside the main content area \*/

}

/\* Styles for individual sections within the main content \*/

section {

    margin-bottom: 40px; /\* Adds space below each section \*/

}

/\* Styles for level 2 headings \*/

h2 {

    text-align: center; /\* Centers the text of the heading \*/

    color: #333; /\* Sets the color of the heading text \*/

}

/\* Styles for the footer section \*/

footer {

    background-color: #333; /\* Dark background color for the footer \*/

    color: #fff; /\* White text color for the footer \*/

    padding: 10px 20px; /\* Adds padding inside the footer \*/

    text-align: center; /\* Centers the footer text \*/

    position: fixed; /\* Fixes the footer at the bottom of the page \*/

    bottom: 0; /\* Positions the footer at the very bottom \*/

    width: 100%; /\* Makes the footer span the entire width of the page \*/

}

/\* Style for tables \*/

table {

    width: 100%; /\* Makes the table span the entire width of its container \*/

    border-collapse: collapse; /\* Collapses table borders into a single border \*/

    margin: 20px 0; /\* Adds margin above and below the table \*/

}

/\* Styles for table headers and cells \*/

th, td {

    border: 1px solid #ddd; /\* Adds a light gray border around cells \*/

    padding: 8px; /\* Adds padding inside table cells \*/

    text-align: center; /\* Centers the text inside table cells \*/

}

/\* Styles for table headers \*/

th {

    background-color: #f2f2f2; /\* Light gray background color for headers \*/

    color: #333; /\* Dark gray text color for headers \*/

}

/\* Styles for even-numbered table rows \*/

tr:nth-child(even) {

    background-color: #f2f2f2; /\* Light gray background color for even rows \*/

}

/\* Styles for table rows on hover \*/

tr:hover {

    background-color: #ddd; /\* Darker gray background color when hovering over a row \*/

}

/\* Styles for the table header section \*/

thead {

    position: sticky; /\* Makes the header stick to the top when scrolling \*/

    top: 0; /\* Positions the sticky header at the top \*/

    background-color: #f2f2f2; /\* Light gray background color for the sticky header \*/

    z-index: 1; /\* Ensures the sticky header is above other content \*/

}

**Topscorer.html**

**Topscorer.html**<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <!-- Title of the webpage -->

    <title>Top Scorers</title>

    <!-- Link to the external CSS file for styling -->

    <link rel="stylesheet" href="Topscore.css">

</head>

<body>

    <!-- Main content section -->

    <main>

        <!-- Section for displaying the Top Scorers table -->

        <section id="top-scorers-section">

            <!-- Heading for the Top Scorers table -->

            <h2>Top Scorers Table</h2>

            <!-- Table to display Top Scorers data, content will be dynamically generated -->

            <table id="top-scorers-table">

                <!-- Top scorers data will be dynamically generated here -->

            </table>

        </section>

    </main>

    <!-- Adding some space before the footer -->

    <br><br><br><br>

    <!-- Footer section with a copyright notice -->

    <footer>

        <p>&copy; 2024 Top Scorers Dashboard</p>

    </footer>

    <!-- Link to the external JavaScript file for functionality -->

    <script src="TopScore.js"></script>

</body>

</html>

**REFERENCES**

**1.** [HTML5 Documentation](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5)

**TESTINGS**

JSON and SCHEMA validated using following website **-https://www.jsonschemavalidator.net/**