

JavaScript ES6 and OOP Part 2 - Assessment

Introduction

This week, you will be doing a project titled **Football Database**. This project will include concepts of ES6 such as **map, filter, find, includes, join, template strings** etc.

1. **find():**

- The **find()** function in JavaScript is a method available for arrays.
- It is used to find the first element in an array that satisfies a specified condition.
- The **find()** method takes a callback function as an argument, and it returns the first element in the array for which the callback function returns true, or undefined if no such element is found.
- **Syntax:** `array.find(callback(element[, index[, array]]), thisArg)`

2. **includes():**

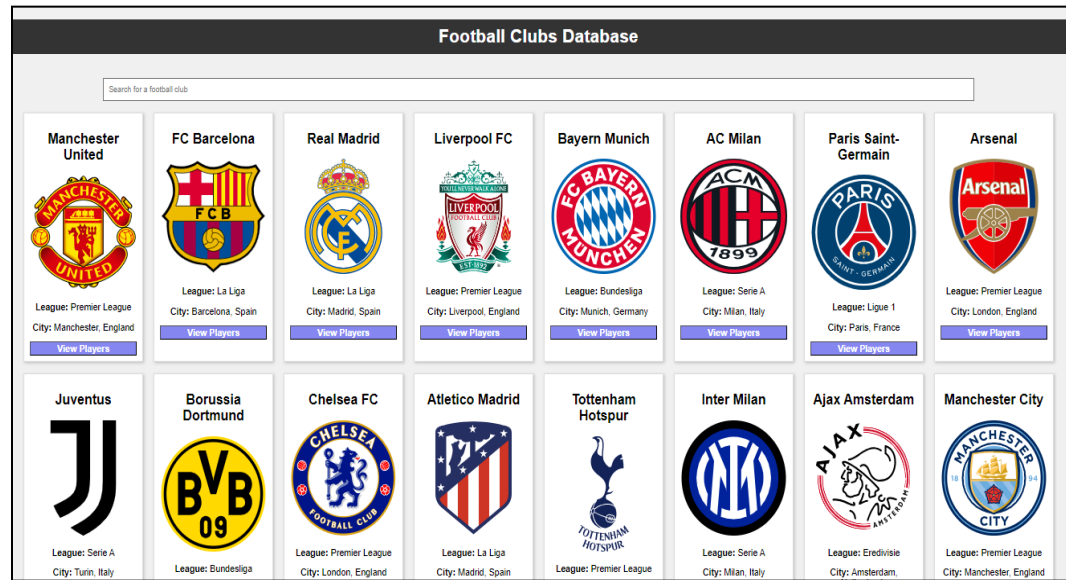
- The **includes()** function in JavaScript is a method available for arrays and strings.
- It is used to check if an array contains a specific element or if a string contains a specified substring.
- The **includes()** method returns true if the element or substring is found in the array or string, and false otherwise.
- **Syntax:** `string.includes(searchString[, position])`

The details of the project are as follows:

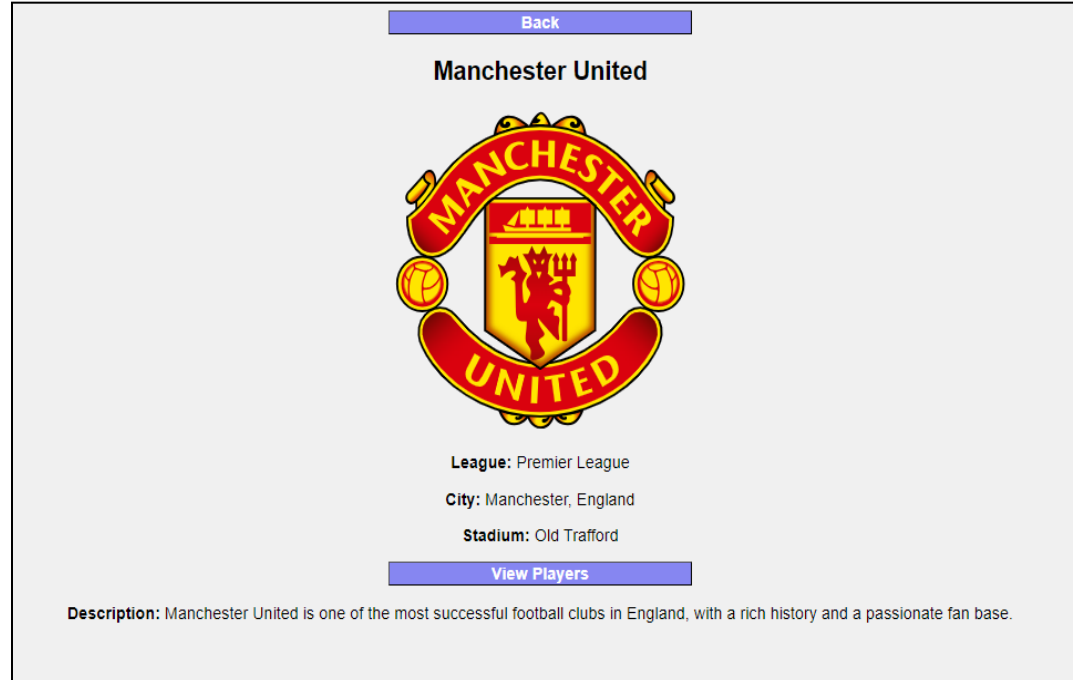
1. The App is divided into following four sections:

1. **Football Club Database Display:** This is the Homepage where all the Football clubs are displayed as individual cards. Each card contains information regarding a club such as **Name of the Club, Logo Image, League** and **City**. Also, there's a button **View Players** on

each card. (This section is already implemented in the source code provided.)



2. **Individual Football Club Details:** When we click on an individual card, it loads the club details such as **Club Title**, **Logo Image**, **League**, **City**, **Stadium** and **Description** on the page. Moreover, two buttons are displayed, **Back** and **View Players**. The functionality of both these buttons is the same as mentioned below in point 3. (You need to implement the logic for this section as mentioned in tasks 1 and 2 in the problem statement below)




3. **View Player Details:** When the **View Players** button is clicked on a particular Football Club card, the details regarding players of that Football Club are displayed. These details include **Player's Name**, **Position**, **Goals and Assists**. Also, there's a **Back** button at the

top which navigates back to the Home page. (You need to implement the logic for this section as mentioned in task 3 in the problem statement below)

Back	
Manchester United Players	
Name: Antony	
Position: Forward	
Goals: 8	
Assists: 22	
Name: Bruno Fernandes	
Position: Midfielder	
Goals: 8	
Assists: 10	
Name: Harry Maguire	
Position: Defender	
Goals: 1	
Assists: 2	

4. **Search Functionality:** There's a Search Box at the top of the Home page which can be used to search a football club on the basis of name, league or city. As soon as we start typing in the Search Box, the filter mechanism starts working and the search results are reflected on the page. (You need to implement the logic for this section as mentioned in task 4 in the problem statement below)

Football Clubs Database	
<input type="text" value="barcelona"/>	
<div><div><div><div>FC Barcelona</div><div></div><div>League: La Liga</div><div>City: Barcelona, Spain</div><div>View Players</div></div></div></div>	

- You will find the following files in the folder:
 - **index.html**: This file contains the html for the web page.
 - **style.css**: This file contains the styling for index.html.
 - **script.js**: In this file, you have to add code to complete the tasks which are mentioned in the Problem Statement.
 - **footballClubs.js**: This file contains the array **footballClubs** which comprises a list of **footballClub** objects. Information about each of the football clubs is fetched from this array.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

1. **FootballClub Card Click**: You have to write code for the **handleClubClick()** function.
 - When a user clicks on a footballClub card, the card element itself should be passed as the argument to the function **handleClubClick()**.
 - The function should then find the footballClub's details from the footballClub database using the club's name and pass it to the **displayClubDetails()** function as argument.
2. **Display Club Details**: You have to write code for the **displayClubDetails()** function.
 - The **displayClubDetails()** function should take the footballClub object as argument.
 - It should display the details of the object such as **Club Name, Logo Image, League, City, Stadium** and **Description** on the webpage.
 - You should also create two buttons: **Back** and **View Players**.
 - The **Back** button should take us back to the Home Page.
 - The **View Players** button should give us the details of the footballClub players.
3. **View Club Players**: You have to write code for the **viewClubPlayers()** function.
 - The **viewClubPlayers()** function takes the club name as argument.
 - It searches the footballClub database to find the details of the club passed as argument.
 - Next, it should display the details of the players of this selected club on the web page.
 - There should be a **Back** button which should lead us back to the Home Page.
4. **Search Football Club**: You have to write code for **handleSearchInput()** function.
 - This function should take no argument.
 - It should find the value of the search box.
 - It should then search the value in the footballClub database on the basis of **club name, club city** or **club league**.
 - The footballClubs which match the criteria should be displayed on the webpage.
 - The **displayClubs()** function should be called to display the filtered footballClubs.

Program Organization

- You will be getting a zip folder containing a folder named **Javascript ES6 and OOP Part 2 Assessment_For Noncoders**.
- The **Javascript ES6 and OOP Part 2 Assessment_For Noncoders** folder has 4 files namely **index.html**, **style.css** and **script.js** and **footballClubs.js**.
- You are required to add functionalities to complete the tasks (stated above in the problem statement) in **script.js** file.

Evaluation Rubric

Total Project Points: 60

- Correctness:
Correctness of implementation
 - Problem statement - point 1 (25%) : 15 Points
 - Problem statement - point 2 (25%) : 15 Points
 - Problem statement - point 3 (25%) : 15 Points
 - Problem statement - point 4 (25%) : 15 Points

Program Instructions

- The **Javascript ES6 and OOP Part 2 Assessment_For Noncoders** folder should have **index.html**, **style.css** and **script.js** and **footballClubs.js** files.
- The **Javascript ES6 and OOP Part 2 Assessment_For Noncoders** folder should be compressed as zip/rar.
- Project will not be evaluated if the submitted project is not in the zip/rar format.