Sports Explorer Documentation

##Overview

Sports Explorer is a simple web application that allows users to explore information about different sports. The application dynamically updates the displayed content based on user interactions with sport buttons.

## How It Works

The application uses HTML, CSS, and JavaScript with the jQuery library. The main components of the code include:

HTML Structure:\*\*

- Defines the structure of the webpage, including buttons for different sports and sections for displaying sport information.

CSS Styling:

- Applies styles to create a visually appealing and responsive layout.

- \*\*JavaScript and jQuery:\*\*

- Utilizes jQuery for event handling and DOM manipulation.

- Defines a sportsData object containing information about various sports.

- Dynamically updates the content in the sportInfo div based on user interactions with sport buttons.

##How it works:

* The website provides users with an entertaining way to learn more about various sports. A user-friendly interface allows users to dynamically update the "sportInfo" section with information about the selected sport by clicking or hovering over sport buttons. For a more engaging experience, each sport has an eye-catching title, an extensive description, and an image. Because of its responsive design, the website is accessible on a range of devices. It offers a user-friendly, visually beautiful interface that makes it simple for consumers to explore and discover a range of sports. When jQuery is integrated, it improves the user experience by enabling fluid interactions and dynamic content updates. Essentially, the website functions as an instructional resource, providing a visually captivating and interactive means for users to learn about

## Goal of the Application

The goal of Sports Explorer is to provide users with a simple and interactive way to learn more about various sports. Users can easily navigate through different sports, gaining insights into each sport's description and visual representation.

##explaining the code

* The Interface design is authentic, by incorporates a simple and modern aesthetic, with a cohesive color scheme and a clear layout.
* jQuery is included in application source: <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
* jQuery is used to bind events to event handlers, the following code snippet shows event handling using jQuery:
  + // Event handler for sport buttons
  + $(".sportButton").on("click mouseover", function () {
  + const sport = $(this).data("sport");
  + const sportData = sportsData[sport];
  + // Update the sportInfo div with the selected sport's information and image
  + $("#sportInfo")
  + .html(`<h2 style="color: #3498db;">${sport}</h2><p>${sportData.description}</p><img id="sportImage" src="${sportData.image}" alt="Sport Image">`);
  + });
    - $(".sportButton") selects all elements with the class "sportButton" (which are the sport buttons in your HTML). .on("click mouseover", function () { ... }) binds a function to both the "click" and "mouseover" events for these elements.
* jQuery to access and manipulate the Document Object Model (DOM).
  + // Update the sportInfo div with the selected sport's information and image
  + $("#sportInfo")
  + .html(`<h2 style="color: #3498db;">${sport}</h2><p>${sportData.description}</p><img id="sportImage" src="${sportData.image}" alt="Sport Image">`);
    - In this code: $("#sportInfo") uses the jQuery selector to select the element with the ID "sportInfo" in the DOM. .html(...) is a jQuery method that sets the HTML content of the selected element.
* dynamic updating of the DOM using jQuery calls.
  + // Update the sportInfo div with the selected sport's information and image
  + $("#sportInfo")
  + .html(`<h2 style="color: #3498db;">${sport}</h2><p>${sportData.description}</p><img id="sportImage" src="${sportData.image}" alt="Sport Image">`);
    - $("#sportInfo") selects the element with the ID "sportInfo" in the DOM. .html(...) is a jQuery method used to set the HTML content of the selected element. The content inside the #sportInfo div is dynamically updated based on the selected sport's information.

##JS annotation (created a separate file for this)

* // Document ready function to ensure script runs after the page is fully loaded
* $(document).ready(function () {
* // Sports data object containing information about various sports
* const sportsData = {
* football: {
* description: "Football is a popular sport played worldwide...",
* image: "football.jpg"
* },
* basketball: {
* description: "Basketball is a fast-paced game played between two teams...",
* image: "basketball.jpg"
* },
* // ... (similar entries for other sports)
* };
* // Event handler for sport buttons, triggered on click or mouseover
* $(".sportButton").on("click mouseover", function () {
* // Retrieve the data-sport attribute from the clicked button
* const sport = $(this).data("sport");
* // Fetch the corresponding data for the selected sport
* const sportData = sportsData[sport];
* // Update the sportInfo div with the selected sport's information and image
* $("#sportInfo")
* .html(`<h2 style="color: #3498db;">${sport}</h2><p>${sportData.description}</p><img id="sportImage" src="${sportData.image}" alt="Sport Image">`);
* });
* });

the provided includes the following:

1. **Events:**
   * The code has an event handler for sport buttons using jQuery. It responds to both "click" and "mouseover" events.
2. **Dynamic Addition of Elements:**
   * The code dynamically adds content to the webpage. When a sport button is clicked or hovered over, the content inside the **#sportInfo** div is dynamically updated based on the selected sport's information.
3. **jQuery Styling (CSS):**
   * jQuery is used to dynamically apply styling. For example, the color of the **<h2>** element inside the **#sportInfo** div is set using inline styles.
4. **Access to the DOM:**
   * The code accesses the DOM using jQuery selectors. For instance, **$(".sportButton")** selects elements with the class "sportButton," and **$("#sportInfo")** selects the element with the ID "sportInfo."