Assignment - 1

**Submission Guidelines:**

* Commit your code in your branch in the repository provided and Include a README.md file that provides description and output of your project.
* Deploy your project to a hosting platform “GitHub Pages” and include the live demo URL in your README.md.(OPTIONAL)
* Include comments in your code to explain the logic and write clean code which is understandable and then commit it.

—---------------------------------------------------------------------------------------------------------

Javascript(Basics,DOM,API)-

Projects - Hangman(Console) and Sports Scoreboard

**Assignment: Console-Based Hangman Game in JavaScript**

**Project Overview:**

In this assignment, you will create a console-based Hangman game using JavaScript. The game will be played exclusively in the browser's JavaScript console, allowing users to guess a hidden word letter by letter.

**Requirements:**

Game Logic:

Choose a list of words or phrases for players to guess. These words should be predefined in your JavaScript code.

Implement a mechanism to randomly select a word from the list for each game.

Display the hidden word as underscores initially, with one underscore for each letter in the word.

Gameplay in Console:

Instruct players to use the browser's console to enter their guesses (letters).

Check if the guessed letter is in the word:

If the letter is in the word, reveal its positions in the hidden word.

If the letter is not in the word, decrement the remaining attempts (lives).

Keep track of guessed letters and display them in the console.

Allow players to keep guessing until they either:

Successfully guess the entire word.

Run out of attempts (lives).

End Game in Console:

Display a message in the console indicating whether the player won or lost.

Allow players to start a new game in the console after the game ends.

**Tips**:

Encourage students to break down the problem into smaller functions to encapsulate different aspects of the game logic.

Provide sample word lists for students to get started, but also encourage them to expand the word list.

Emphasize the importance of user input validation and error handling in the console.

Encourage creativity in designing win/loss messages and user interaction within the console.

—---------------------------------------------------------------------------------------------------------

**Assignment: Real-Time Sports Scoreboard Project**

**Project Overview:**

In this assignment, you will create a real-time sports scoreboard web application that fetches live sports scores from a sports data API and updates the scores in real-time. The project will involve making API requests, handling JSON data, and updating the scoreboard dynamically using JavaScript.

**Requirements**:

API Integration:

Choose a sports data API (e.g., a free API like the "Sports Open Data" API or a commercial API).

Fetch live sports scores from the API.

Scoreboard Display:

Create a simple HTML layout for the scoreboard, including team names and scores.

Use CSS to style the scoreboard and make it visually appealing.

Real-Time Updates:

Implement real-time updates by periodically making API requests (e.g., every 30 seconds) to fetch the latest scores.

Update the scoreboard dynamically with the new scores without refreshing the page.

Team Names:

Display the names of the competing teams.

Error Handling:

Implement error handling to deal with API request failures gracefully.

Display an error message to the user in case of API errors.

Additional Features (Optional):

Display additional information, such as match status, date, and time.

Add a feature to allow users to choose different sports or matches to follow.

Implement user-friendly features like loading spinners during API requests.

**Tips**:

* Explore different sports data APIs and choose one that suits their project.
* Use asynchronous JavaScript (e.g., fetch or XMLHttpRequest) for API requests.
* Emphasize the importance of error handling to ensure a smooth user experience.
* Encourage creativity in styling and optional feature implementation.

—---------------------------------------------------------------------------------------------------------