|  |  |
| --- | --- |
| Name :Ross Paolo Villavicencio | Section : BSIT 3A |
| Assignment No : 3 | Submission Date : Apr. 6, 2024 |
| Assignment Title: Guessing Number Game | |

**Code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Number Guessing Game</title>

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

</head>

<body>

<div class="container">

<h1>Number Guessing Game</h1>

<p>Guess a number between 1 and 10:</p>

<input type="text" id="userGuess">

<button onclick="checkGuess()">Submit</button>

<p id="message"></p>

</div>

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

</body>

</html>

**JS FILE**

const randomNumber = Math.floor(Math.random() \* 10) + 1;

let attempts = 0;

function checkGuess() {

const userGuess = parseInt(document.getElementById("userGuess").value);

const messageElement = document.getElementById("message");

if (isNaN(userGuess) || userGuess < 1 || userGuess > 10) {

messageElement.textContent = "Invalid input: Please input a number between 1 and 10.";

return;

}

attempts++;

if (userGuess === randomNumber) {

messageElement.textContent = `Congratulations! You've guessed the correct number (${randomNumber}) in ${attempts} attempts.`;

} else if (userGuess < randomNumber) {

messageElement.textContent = "Too low! Try again.";

} else {

messageElement.textContent = "Too high! Try again.";

}

}

**CSS FILE**

body {

font-family: Arial, sans-serif;

}

.container {

text-align: center;

margin-top: 100px;

}

input[type="text"] {

padding: 8px;

}

button {

padding: 8px 16px;

margin-top: 8px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

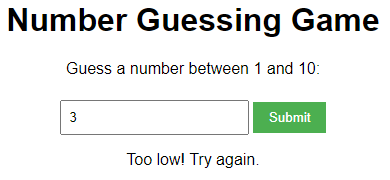
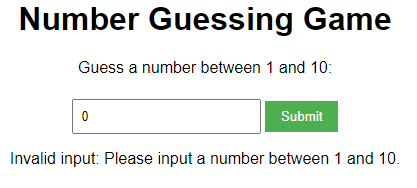
}

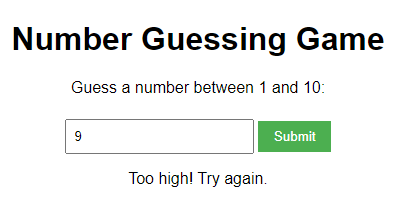
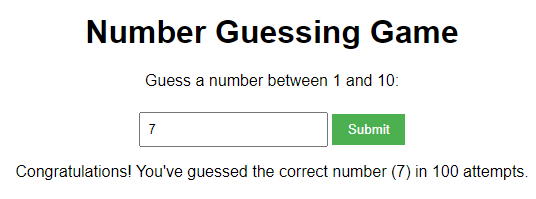
button:hover {

background-color: #45a049;

}

**SCREENSHOT**





**Dictionary:**

Dictionary:

1. `alert` - method displays an alert box with a specified message and an OK button.

- Syntax: `alert(message)`

- Parameter: `message` - a string that you want to display in the alert box.

- Example:

alert('Hello, World!');

- Output: An alert box with the message 'Hello, World!'.

2. `prompt` - method displays a dialog box that prompts the visitor for input.

- Syntax: `prompt(text, defaultText)`

- Parameter: `text` - a string of text to display to the user. `defaultText` - a default value for the input field (optional).

- Example:

const userInput = prompt('Please enter your name', 'Harry Potter');

- Output: A dialog box with the prompt 'Please enter your name' and a prefilled answer 'Harry Potter'.

3. `parseInt` - function parses a string argument and returns an integer of the specified radix (the base in mathematical numeral systems).

- Syntax: `parseInt(string, radix)`

- Parameter: `string` - the value to parse. `radix` - an integer between 2 and 36 that represents the base of the number in the string.

- Example:

const number = parseInt('10', 10);

- Output: 10

4. `Math.random` - function returns a floating-point, pseudo-random number between 0 (inclusive) and 1 (exclusive).

- Syntax: `Math.random()`

- Parameter: None.

- Example:

const randomNum = Math.random();

- Output: A number between 0 and 1, such as `0.5`.

5. `while` - statement creates a loop that executes a block of code as long as the test condition evaluates to true.

- Syntax: `while (condition) { // code block to be executed }`

- Parameter: `condition` - an expression evaluated before each pass through the loop. If this condition evaluates to true, the block of code is executed.

Example:

let count = 1;

while (count < 4) {