|  |  |
| --- | --- |
| Name : John Philip G. Cauilan | Section : BSIT 3A |
| Activity No : 4 | Submission Date : May 4, 2024 |
| Activity Title: Fibonacci Sequence | |

**Code :**

**HTML FILE**

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

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

*<link rel="stylesheet" href="styles.css">*

*<title>Fibonacci Series Generator</title>*

*</head>*

*<body>*

*<main>*

*<h1>Fibonacci Series Generator</h1>*

*<label for="num-terms">Number of terms:</label>*

*<input type="number" id="num-terms" min="1" max="99" placeholder="Enter a number between 1 and 99">*

*<button id="generate-series">Generate Series</button>*

*<div id="error-message" class="hidden">Please enter a valid number between 1 and 99.</div>*

*<div id="resultDiv"></div>*

*</main>*

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

*</body>*

*</html>*

**JS FILE**

*const numTermsInput = document.getElementById('num-terms');*

*const generateSeriesButton = document.getElementById('generate-series');*

*const resultDiv = document.getElementById('resultDiv');*

*const errorMessage = document.getElementById('error-message');*

*generateSeriesButton.addEventListener('click', () => {*

*const numTerms = parseInt(numTermsInput.value.trim(), 10);*

*if (isNaN(numTerms) || numTerms <= 0 || numTerms >= 100) {*

*resultDiv.textContent = '';*

*errorMessage.textContent = 'Ensure the number is not greater than or equal to 100.';*

*errorMessage.classList.remove('hidden');*

*generateSeriesButton.disabled = true;*

*setTimeout(() => {*

*generateSeriesButton.disabled = false;*

*}, 1000);*

*} else {*

*errorMessage.textContent = '';*

*errorMessage.classList.add('hidden');*

*const fibonacciSeries = [0, 1];*

*for (let i = 2; i < numTerms; i++) {*

*fibonacciSeries[i] = fibonacciSeries[i - 1] + fibonacciSeries[i - 2];*

*}*

*resultDiv.textContent = fibonacciSeries.slice(0, numTerms).join(', ');*

*window.scrollTo({ top: 0, behavior: 'smooth' });*

*generateSeriesButton.disabled = true;*

*setTimeout(() => {*

*generateSeriesButton.disabled = false;*

*}, 1000);*

*}*

*});*

**CSS FILE**

*body {*

*font-family: 'Roboto', sans-serif;*

*background-color: #f0f0f0;*

*display: flex;*

*justify-content: center;*

*align-items: center;*

*height: 100vh;*

*margin: 0;*

*overflow: hidden;*

*}*

*main {*

*background-color: #fff;*

*padding: 20px;*

*border-radius: 5px;*

*box-shadow: 0 0 10px 2px rgba(0, 0, 0, 0.1);*

*width: 100%;*

*max-width: 500px;*

*text-align: center;*

*position: relative;*

*overflow: hidden;*

*}*

*h1 {*

*margin-top: 0;*

*font-size: 2rem;*

*color: #4CAF50;*

*text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);*

*}*

*input[type="number"] {*

*display: block;*

*width: 100%;*

*padding: 10px;*

*border: 1px solid #ccc;*

*border-radius: 5px;*

*font-size: 1.2rem;*

*margin-bottom: 10px;*

*box-sizing: border-box;*

*transition: border-color 0.3s ease;*

*}*

*input[type="number"]:focus {*

*border-color: #4CAF50;*

*}*

*button {*

*background-color: #6CA0DC;*

*color: #fff;*

*padding: 10px 20px;*

*border: none;*

*border-radius: 25px;*

*cursor: pointer;*

*transition: background-color 0.3s ease;*

*font-size: 1.2rem;*

*margin-top: 10px;*

*display: inline-block;*

*margin-bottom: 10px;*

*position: relative;*

*overflow: hidden;*

*}*

*button:hover {*

*background-color: #4CAF50;*

*}*

*button:active {*

*transform: translateY(2px);*

*}*

*button::before {*

*content: '';*

*position: absolute;*

*top: 0;*

*left: 0;*

*width: 100%;*

*height: 100%;*

*background-color: rgba(255, 255, 255, 0.2);*

*z-index: -1;*

*transition: transform 0.3s ease;*

*}*

*button:hover::before {*

*transform: translateX(-100%);*

*}*

*.result-section {*

*margin-top: 20px;*

*}*

*.result-text {*

*font-size: 1.2rem;*

*font-weight: bold;*

*margin-top: 10px;*

*color: #4CAF50;*

*text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);*

*}*

*.result-text::selection {*

*background-color: #6CA0DC;*

*color: #fff;*

*}*

*.hidden {*

*display: none;*

*}*

*.error-message {*

*color: red;*

*font-size: 0.8rem;*

*margin-top: 5px;*

*}*

*@media (max-width: 600px) {*

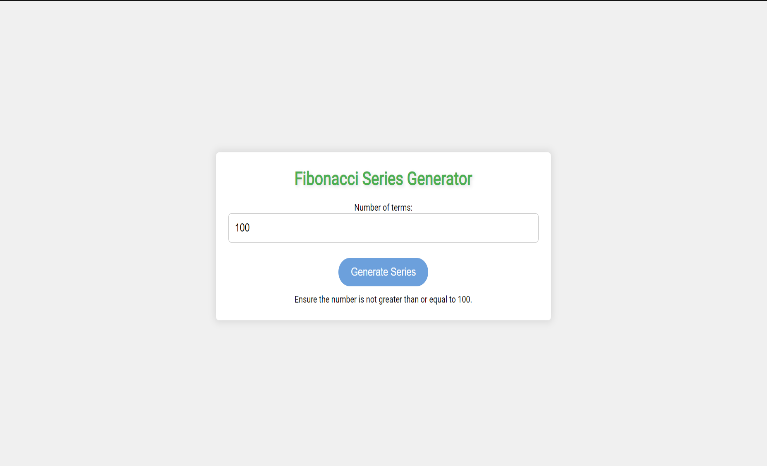
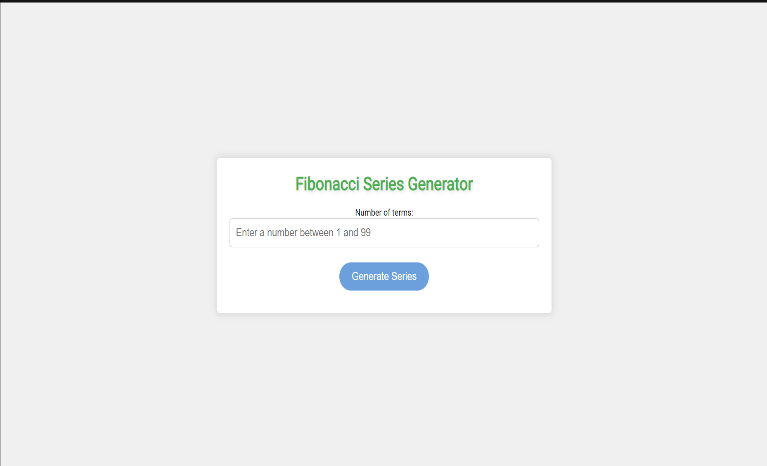
*main {*

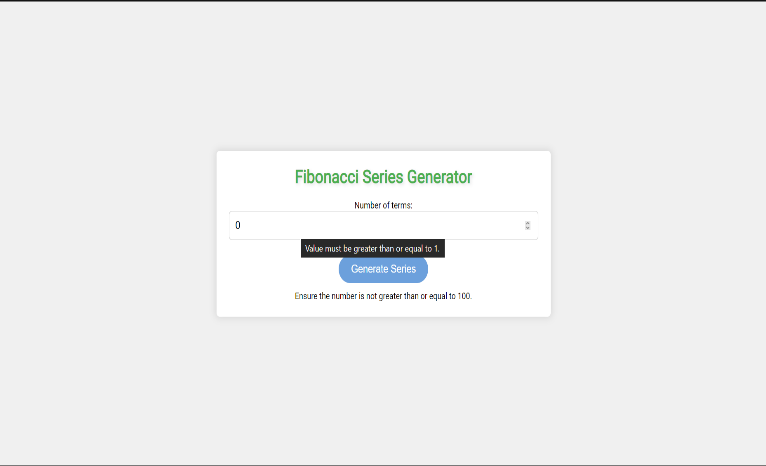
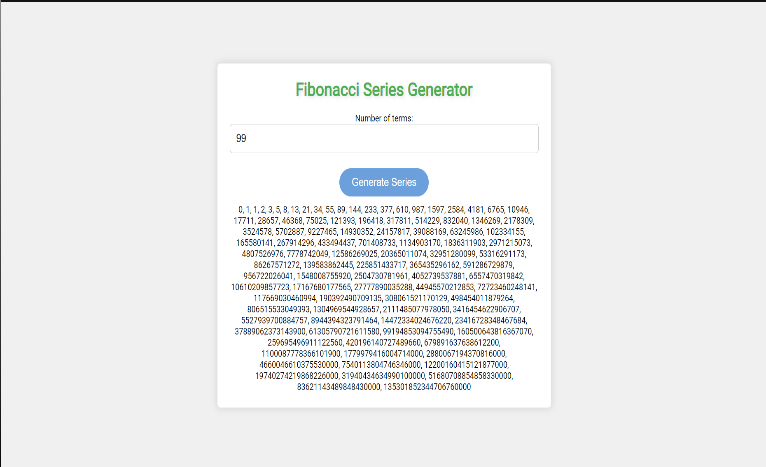
*max-width: 100%;*

*}*

*}*

**SCREENSHOT**

******

**

**Dictionary:**

*// List keywords, tags, or methods that are new to you and explain how they are used in the code.*

*// atleast 5 tag, methods or keyword*

* **addEventListener() method**: The addEventListener() method attaches an event handler function to an HTML element.  
  **Syntax**: generateSeriesButton.addEventListener('click', () => {

const numTerms = parseInt(numTermsInput.value.trim(), 10);

* **Ensure**: This term means "make sure". It is used to provide a precondition that the number of terms should not be greater than or equal to 100. **Syntax**: errorMessage.textContent = 'Ensure the number is not greater than or equal to 100.';

* **Array slice method**: The slice() method returns a shallow copy of a portion of an array into a new array object.

**Syntax**: resultDiv.textContent = fibonacciSeries.slice(0, numTerms).join(', ');

* **setTimeout() method**: The setTimeout() method sets a timer that executes a function or specified piece of code once the timer expires.

**Syntax**: setTimeout(() => {

generateSeriesButton.disabled = false;

}, 1000);

**Parameter**: => {

generateSeriesButton.disabled = false;

}, 1000);

* **window.scrollTo() method:** The scrollTo() method scrolls to a particular set of coordinates in the document.

**Syntax**: window.scrollTo({ top: 0, behavior: 'smooth' });

**Parameter** ({ top: 0, behavior: 'smooth' })