**З В І Т**

з лабораторного заняття № 2 дисципліни

«Програмні та інструментальні засоби веб-дизайну»

студента гр. КІ-21-2

Шейко Р. О.

**Завдання 1.1**

Засобами HTML, CSS3 і JavaScript створити прикладний web-додаток, з блоком, призначеним для обчислення натурального логарифму будь якого цілого, або дійсного числа. Блок повинен містити заголовок «Обчислення натурального логарифму числа X» і кнопки з написами «Увести число X» і «Обчислити logX». Кнопки відділити від заголовку горизонтальною лінією. Уведення значення числа в скрипт забезпечити за допомогою методу prompt() об’єкту window. У скрипті, що запускається за подією click при натисканні кнопки «Увести число X», забезпечити передачу значення числа X відповідній змінній скрипта за допомогою методу prompt(), а також перевірку заповнення користувачем текстове поля форми вікна і відповідність уведеного значення числа X цілому або дійсному типу. У випадку коли рядок уведений до поля форми пустий або не відповідає числовому значенню, методом alert() об’єкту window повинно відкриватися вікно з повідомленням «Уведіть ціле або дійсне число в поле форми вікна». Перевірку на предмет заповнення поля можна здійснити порівнянням уведеного значення X з пустим рядком "", а на відповідність X числовому значенню за допомогою функції верхнього рівня isNaN(). Окрім цього забезпечити перевірку уведеного числа X на нуль і від’ємність. Коли X дорівнює 0 або від’ємне (логарифму таких чисел не існує), повинно методом alert() відкриватися вікно з повідомленням «Уведіть не нульове і не від'ємне число».

При натисканні кнопки «Обчислити logX» за подією click повинен запускатися скрипт, який використовуючи відповідний метод об’єкту Math, обчислює значення натурального логарифму числа X и виводить його у вікно, відкрите методом alert(), як напис «logX= значення логарифму». У значенні логарифму, за допомогою функції toFixed() забезпечити 3 цифри після десяткової крапки. Для запобігання невизначеності розрахунку logX, коли користувач натискає кнопку «Обчислити logX» до уведення значення числа X, у скрипті, де виконується розрахунок logX, можна використати умовний оператор з умовою x==undefined. При виконанні цієї умови методом alert() відкрити вікно з повідомленням «Уведіть число X». Забезпечити підсвітку кнопок при наведенні курсору миші шляхом зміни фонового кольору і кольору напису відповідними властивостями CSS для анімаційних ефектів. Забезпечити графічний фон web- сторінки доступним вам графічним файлом. Фонове зображення не повинно повторюватися.

Можливій дизайн web- сторінки ілюструють скриншоти, наведені на рис. 1 і рис. 2. На рис. 3 наведені фрагменти сторінок з вікнами, що відкриваються при різних ситуація на етапах уведення числа X і обчислення його натурального логарифму.

|  |
| --- |
|  |
| Рис. 1. Можливий вихідний вигляд web- сторінки додатку для обчислення  натурального логарифму будь яких цілих і дійсних чисел |

|  |
| --- |
|  |
| Рис. 2. Можливий вигляд web- сторінки додатку для обчислення натурального логарифму числа 236.58 на етапі уведення числа. Натиснута кнопка «Увести число X» |

|  |  |
| --- | --- |
|  |  |
| **а** | **б** |
|  |  |
| **в** | **г** |
|  | |
| **д** | |
| Рис. 3. Вікна, що відкриваються при різних ситуація на етапах уведення числа X і обчислення його натурального логарифму: а – уводиться число 236.58; б – уводиться число 236,58 і натискається «OK»; в – уводиться 0 або будь яке від’ємне число; г – коли користувач передчасно натиснув кнопку «Обчислити logX» до уведення значення числа X; д – коли було уведено число 236.58 і користувач натиснув кнопку «Обчислити logX» | |

**HTML код. Файл «index.html»**

<!DOCTYPE html>

<html>

<head>

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

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

</head>

<body>

<center>

<div class="block">

<div class = "zag">ќбчисленн¤ натурального логарифму числа X</div>

<hr>

<button onClick="vvod()">”вести число X</button><br>

<button onClick="log()">ќбчислити logX</button><br>

<hr>

</div>

</center>

</body>

</html>

**CSS код. Файл «style.css»**

body

{

background: url(https://oir.mobi/uploads/posts/2021-03/1616548867\_4-p-matematicheskii-fon-5.jpg);

background-size: 100%;

font-family: Comic Sans MS, sans-serif;

}

.block

{

text-align: center;

background: #488091;

border: dashed 7px #1E4482;

border-radius:70px;

width: 900px;

height: 370px;

}

.zag

{

color: white;

font-size: 9mm;

font-weight: 900px;

}

button

{

width: 650px;

height: 120px;

border: dashed 3px #1E4482;

border-radius:70px;

background-color: #314452;

color:white;

font-size:45px;

transition-property:color, background-color:;

font-family: Comic Sans MS, sans-serif;

}

button:hover

{

color:black;

background-color:white;

}

hr

{

height:1mm;

background-color:white;

}

**JS код. Файл «script.js»**

let x;

function vvod()

{

x=prompt("¬вед≥ть значенн¤ X до форми");

if(x == ''){

alert("”вед≥ть ц≥ле або д≥йсне число в поле форми в≥кна.");}

else if(isNaN(x) != 0){

alert("”вед≥ть ц≥ле або д≥йсне число в поле форми в≥кна.");}

else if(Math.sign(x) == 0 || Math.sign(x) == -1 || Math.sign(x) == -0){

alert("”вед≥ть не нульове ≥ не в≥д'Їмне число.")}

}

function log()

{

if(x == undefined){

alert("”вед≥ть число X.");}

else if(isNaN(x) == 0){

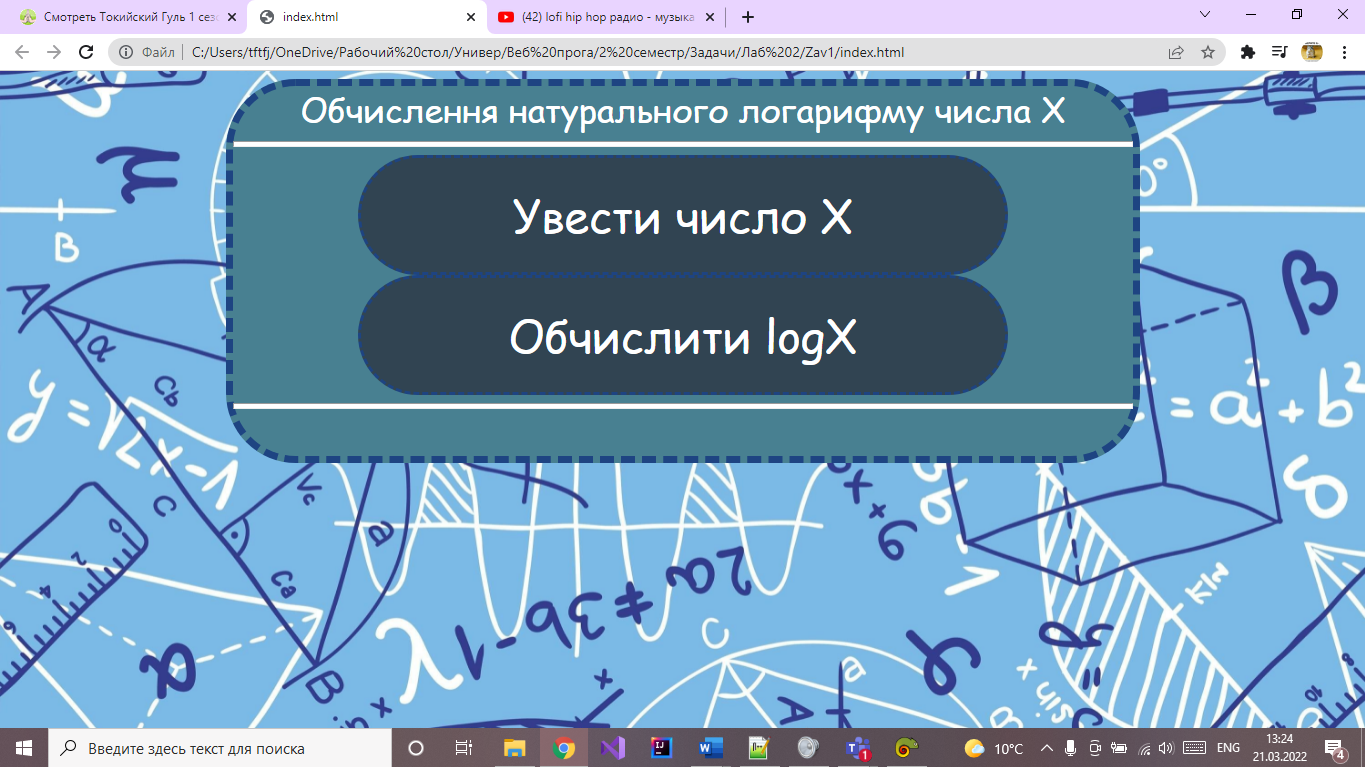
alert("logX = "+Math.log(x).toFixed(3));}

else if(isNaN(x) != 0){

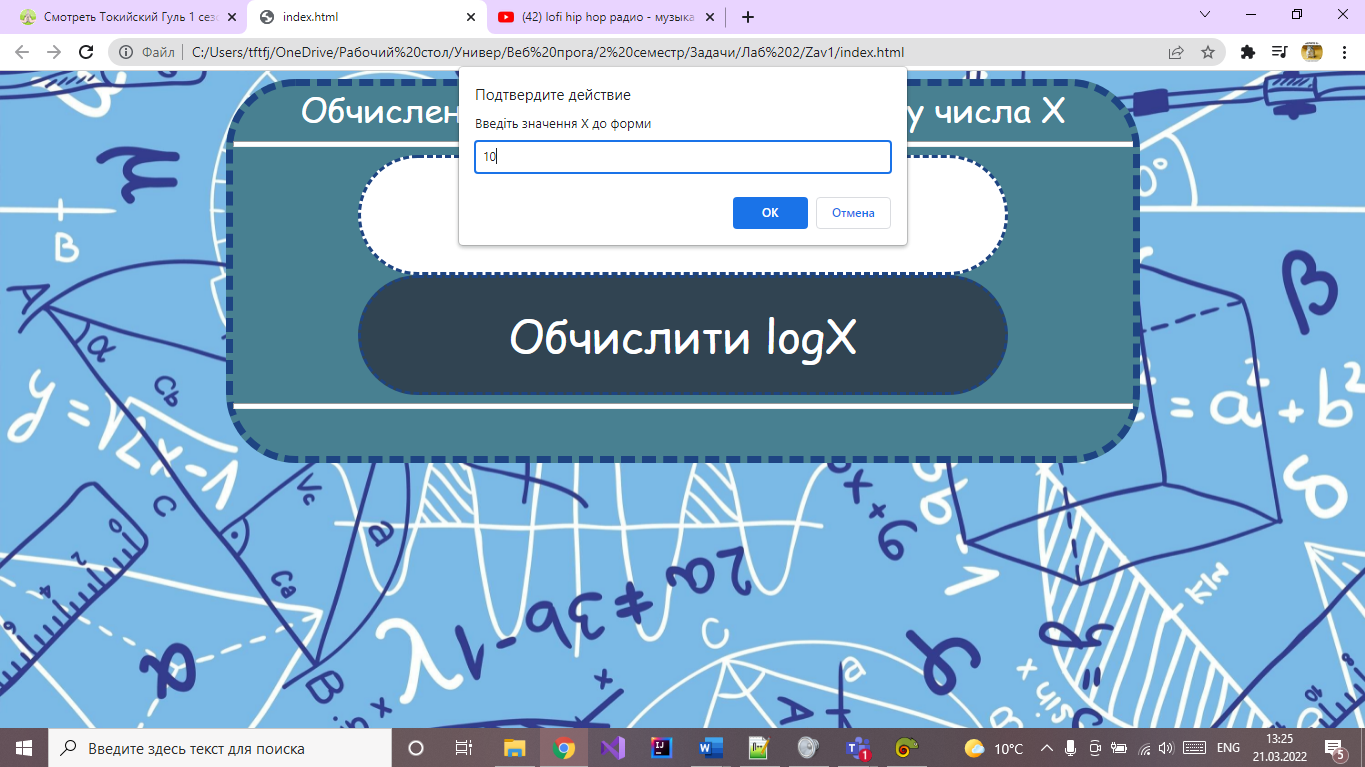
alert("”вед≥ть ц≥ле або д≥йсне число в поле форми в≥кна.");}

}

**Скріншоти, що ілюструють роботу веб-сторінки**



|  |
| --- |
|  |
| Рис. 1. Вихідний вигляд веб-сторінки додатку |
|  |



|  |
| --- |
| Рис. 2. Вигляд веб-сторінки додатку, коли юзер навів та клацнув до кнопки «Увести число Х», ввів число 10 |
|  |

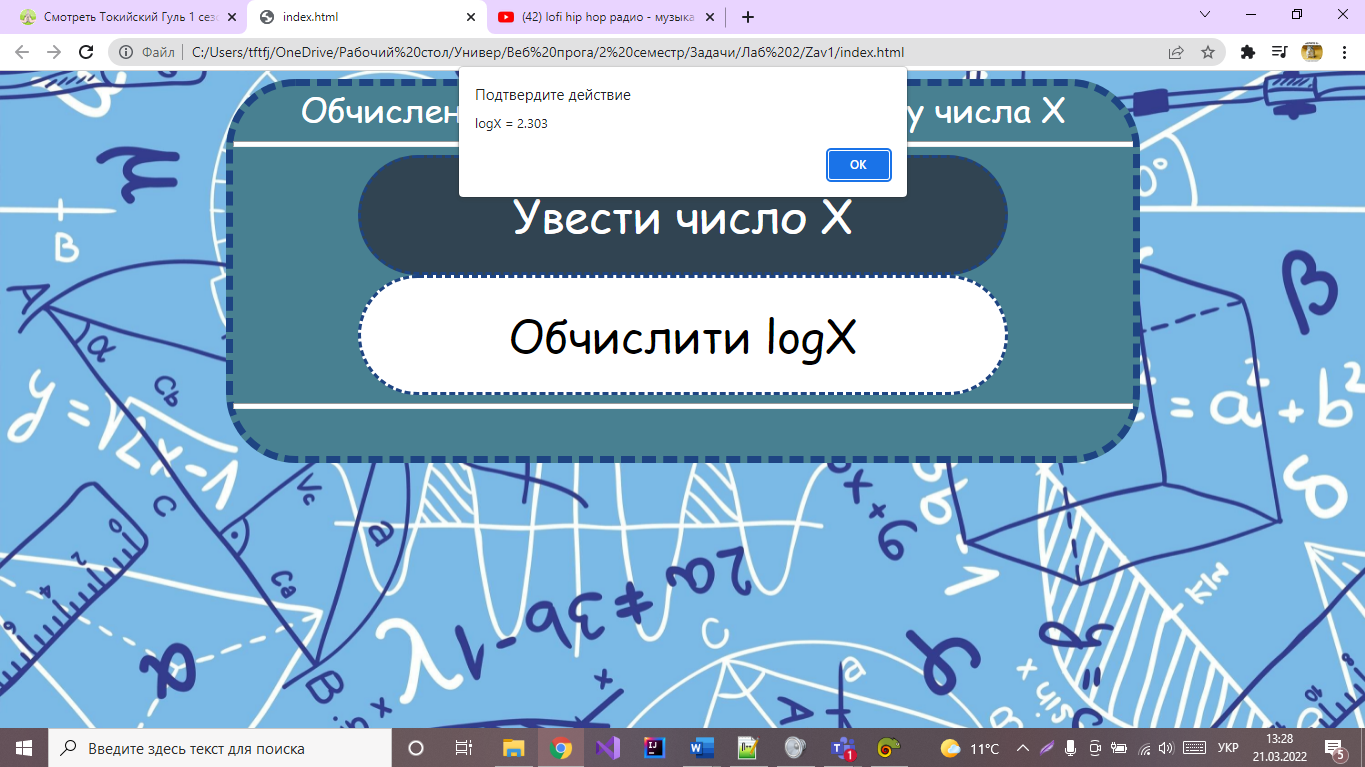


Рис. 3. Вигляд веб-сторінки додатку, коли юзер навів та клацнув до кнопки «Обчислити logX»,та перед цим ввів число 10

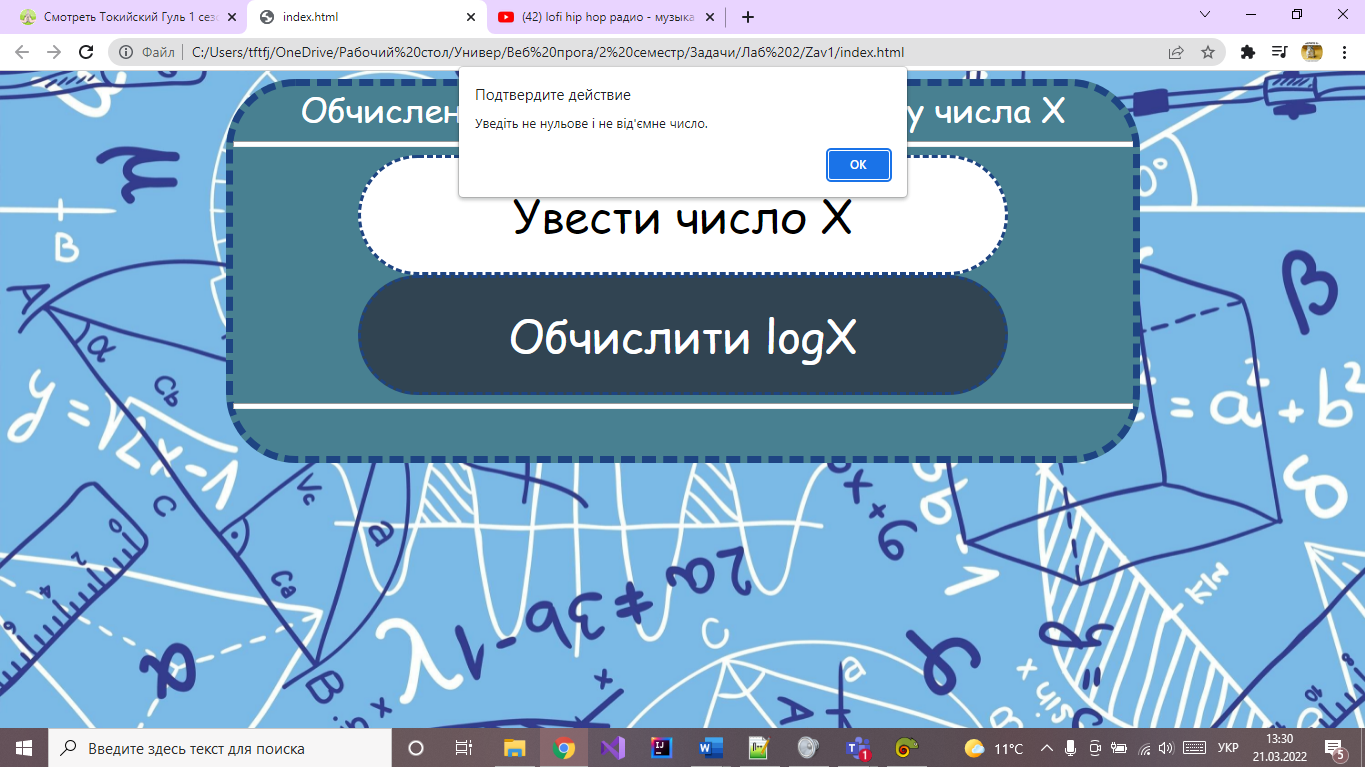


Рис. 4. Вигляд веб-сторінки додатку, коли юзер навів та клацнув до кнопки «Увести число Х», ввів число 0

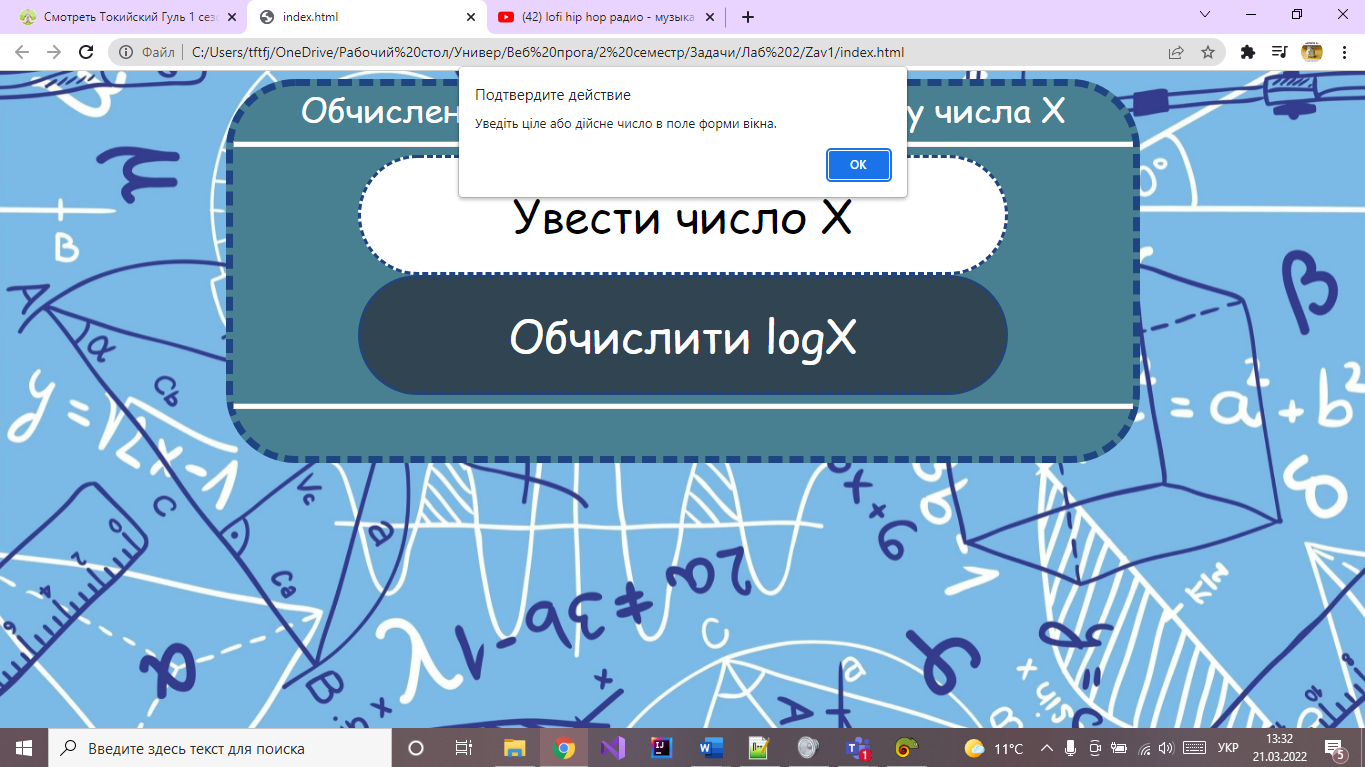


Рис. 5. Вигляд веб-сторінки додатку, коли юзер навів та клацнув до кнопки «Увести число Х», ввів число 0,0008

**Завдання 1.2**.

Засобами HTML, CSS3 і JavaScript створити прикладний web-додаток, з блоком, призначеним для генерації випадкового цілого числа в діапазоні від 0 до 1000. Блок повинен містити заголовок «Генератор випадкових цілих чисел» і відділену від нього горизонтальною лінією кнопку з написом «Виконати генерацію». Рекомендовані розміри блоку 500×170 пікселів, кнопки 495×60 пікселів. Заголовок вивести курсивом Arial кольору з RGB #800000 і висотою літер 9 мм. Фон блоку, кнопки і колір напису на кнопці на ваше розуміння. Забезпечити підсвітку кнопок при наведенні курсору миші шляхом зміни фонового кольору і кольору напису відповідними властивостями CSS для анімаційних ефектів. Блок повинен мати жовту рамку стилю groove товщиною 2 мм. Позиціонувати блок по центру веб-сторінки, забезпечивши таке позиціонування не залежно від розмірів монітору комп’ютера. Фон сторінки задати RGB кодом aabbcc.

При натисканні користувачем на кнопку «Виконати генерацію» повинен запускатися скрипт, який методом random() об’єкту Math генерує випадкове число перетворює його в ціле число, яке лежить в діапазоні 0 – 1000 і виводить його значення в діалогове вікно, відкрите методом alert(), у вигляді напису «Випадкове число = значення числа». Для перетворення дійсного числа в ціле використати функцію верхнього рівня мови JavaScript toFixed().

Можливий дизайн веб-сторінки застосунку і її стан після натискання кнопки «Виконати генерацію» показано на рис. 4.

|  |
| --- |
|  |
| Рис. 4. Вигляд веб-сторінки прикладного web- застосунку для генерації випадкових цілих чисел після натискання кнопки «Виконати генерацію» |

**HTML код. Файл «index.html»**

<!DOCTYPE html>

<html>

<head>

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

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

</head>

<body>

<div class="block">

<div class = "txt">Генератор випадкових цілих чисел</div>

<hr/>

<button onClick="random()">Виконати генерацію</button>

<hr/>

</div>

</body>

</html>

**CSS код. Файл «style.css»**

body

{

background: #A8A84F;

}

.block

{

position: absolute;

top:50%;

left:50%;

margin-left: -250px;

margin-top: -85px;

text-align: center;

background: #BE7839;

border: groove 2mm yellow;

width: 500px;

height: 170px;

}

.txt

{

color: #800000;

font-size: 9mm;

font-family: Arial;

font-style: italic;

}

button

{

width: 495px;

height: 60px;

background-color: #696969;

color:white;

font-size:45px;

transition-property:color, background-color:;

}

button:hover

{

color:black;

background-color:white;

}

hr

{

height:1mm;

background-color:#A8A84F;

}

**JS код. Файл «script.js»**

let min=1;

let max=1000;

function random() {

let result = Math.random() \* (max - min) + min;

alert("Випадкове число = "+result.toFixed(0));

}

**Скріншоти, що ілюструють роботу веб-сторінки**

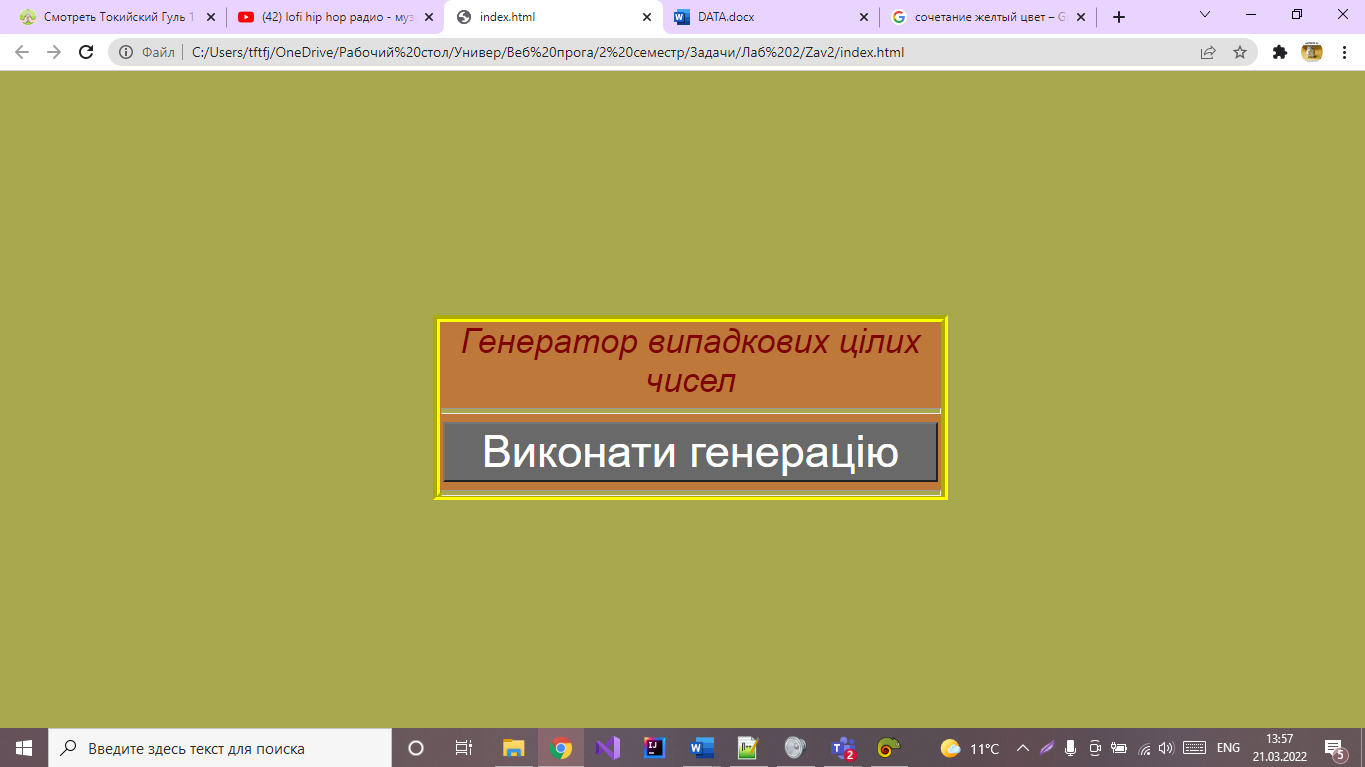


Рис. 1. Вихідний вигляд веб-сторінки додатку

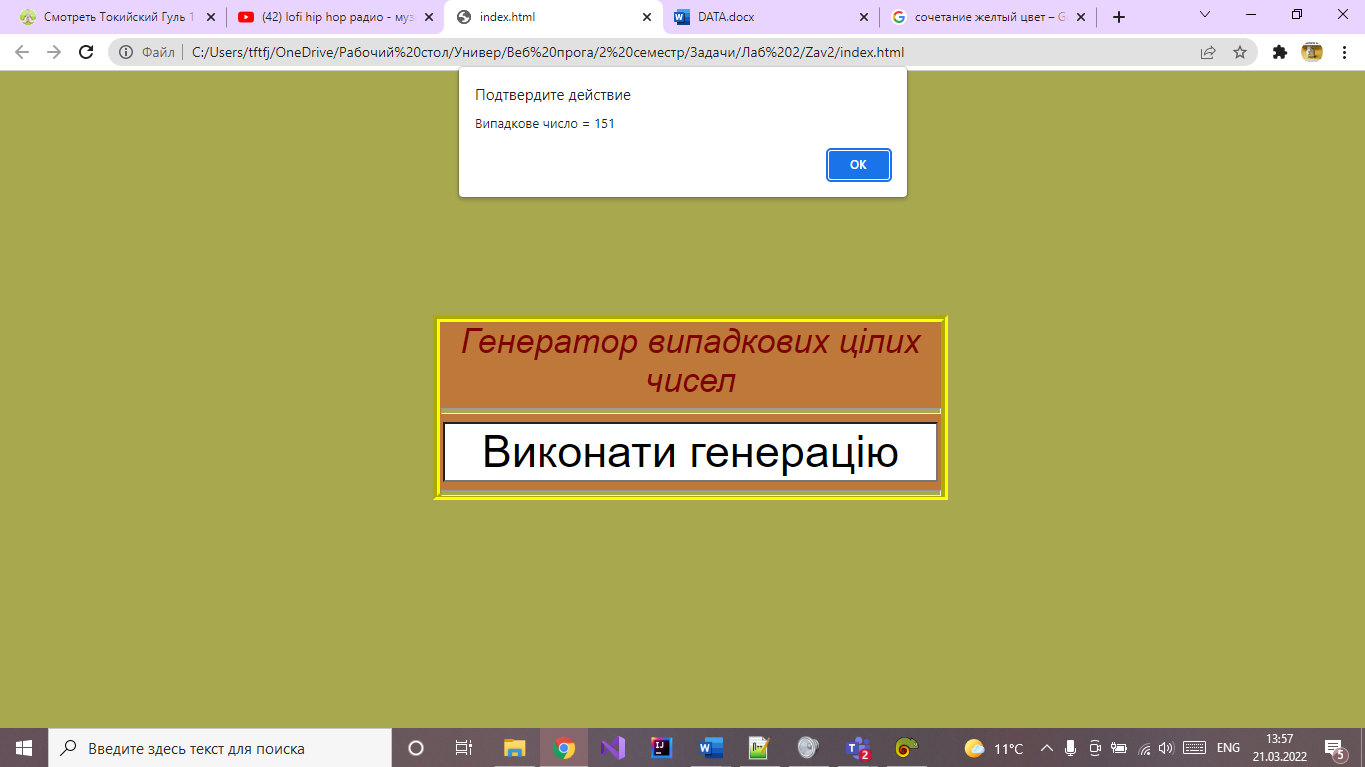


Рис. 2. Вигляд веб-сторінки додатку, коли користувач навів курсор до “Виконати генерацію”