

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №3**

**з дисципліни «Інтернет-технології»**

**на тему:**

**«**JavaScript»

Виконав:

студент гр. ПЗ2011

Проценко Р.О.

Прийняв:

Макаров О. В.

Дніпро, 2023

**Тема.** JavaScript.

**Теоретичні відомості**

Мова JavaScript за короткий час змогла стати однією з основних мов програмування для веб. За допомогою нього можна як модернізувати існуючий сайт на HTML, так і створити новий Web-додаток. Хоча JavaScript не настільки популярний як PHP, але, тим не менше менш, дає деякі можливості, не властиві інших мов програмування. Наприклад, мова JavaScript дозволяє поєднувати різні web-технології, причому ефективно.

Багато хто вважає JavaScript клієнтською мовою, що справді є правдою. У мові багато уваги надається впровадженню в HTML для безпроблемної роботи в середовищі, яке підтримують сучасні браузери. Крім цього, JavaScript є серверною мовою сценаріїв. У цьому плані першими, хто використовував JS, були програмні рішення від Microsoft і Netscape.

Мова дозволяє вирішувати завдання різного рівня – його використовують web-дизайнери, Java-програмісти, розробники БД та звичайні користувачі. Перед початком вивчення JavaScript рекомендується вивчити базові веб-технології. Навички програмування вітаються, але не є необхідними.

Для повного розуміння JavaScript необхідно прояснити мету створення мови. Адже його можливості вже давно переступили межі звичайного інтернет-браузера. Його використовують на серверах, також він убудований в останні версії Windows. Ця мова є потужною інструментом для взаємодії різних технологій і в результаті – створення продуманих та ефективних програмних рішень

Необхідно розуміти, що JavaScript не є інструментом для розробки, він мова сценаріїв. Виконання коду JavaScript є результатом спільної роботи як браузера клієнта, і серверного ПО. На відміну від HTML, JavaScript повністю залежить від браузера, якщо останній не підтримує його, код JavaScript не здійсниться. У цьому випадку код сценарію навіть може вивестись як текст. Оскільки JavaScript є мовою сценаріїв, його можна використовувати у досить різних ситуаціях. Але в більшості випадків, основне застосування припадати на клієнтську частину.

Початкову версію JavaScript розробила Netscape (ще до розробки Java), для покращення HTML - на той момент йому не вистачало динаміки. Після появи мови Java, Netscape стала співпрацювати з компанією Sun(розробник Java) для створення мови сценаріїв з синтаксисом схожим Java. В результаті цієї співпраці народилася мова JavaScript. Він дозволив чітко розділити обов'язки клієнтської та серверної частини. До цього серверна частина виконувала всі операції, аж до контролю за введенням даних, що було досить неефективно.

Використання JavaScript значно збільшило роль клієнтської частини, що дійсно дозволило створити клієнт-серверну систему з його основі. Також, JavaScript доповнив можливості Java, який дозволяв з'єднувати кілька технологій на web-сторінці.

Після появи в 1995 р., мова JavaScript отримала підтримку багатьох відомих розробників програмного забезпечення. За допомогою їх підтримки мова удосконалюється та використовується не тільки в браузерах та різних додатках. Через деякий час права на мову викупила Microsoft, щоправда, здобувши права лише на технологію. В результаті втручання Microsoft було створено свій варіант JavaScript, який підтримував Internet Explorer, але несумісний з Netscape Navigator 3.0, який у свою чергу підтримував свої версії JS. З цього моменту почалася епоха несумісних версій – та сама сторінка по-різному відображалася у різних браузерах. Внаслідок цієї боротьби Netscape пішла з ринку браузерів. Також була розроблена альтернатива JavaScript – VBScript для спрощення розробки на Visual Basic Проте JavaScript став стандартом для Web.

**Завдання**

Додати до сайту, створеного в лабораторній роботі №2 JavaScript функціонал.

**HTML код сторінки**

<!DOCTYPE html>

<html lang="uk">

<head>

    <meta charset="UTF-8">

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

    <title>Класифікація зображень за допомогою нейронних мереж</title>

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

    <!-- <script src="functional.js"></script> -->

</head>

<body>

    <header>

        <h1>Класифікація зображень за допомогою нейронних мереж</h1>

    </header>

    <section>

        <h2>Що таке класифікація?</h2>

        <p>Класифікація - це процес визначення класу або категорії, до якої належить вхідний об'єкт або дані, наприклад

            зображення.

            Це одна з основних завдань машинного навчання та обробки зображень, де нейронні мережі проявили

            ефективність.

            Нейронні мережі для класифікації використовуються в різних галузях, включаючи комп'ютерне зорове

            розпізнавання,

            медицину, автомобільну промисловість та інше.</p>

    </section>

    <section >

        <div id="dataset-container" class="center">

            <!-- Інформація про датасет датасет-->

        </div>

        <div class="nav-button-container">

            <button id="prev-btn" class="nav-button">&#9664;</button> <!-- Символ трикутника вліво -->

            <button id="next-btn" class="nav-button">&#9654;</button> <!-- Символ трикутника вправо -->

        </div>

    </section>

    <section>

        <h2>Популярні моделі для класифікації</h2>

        <ul>

            <li>ResNet (Residual Network)</li>

            <li>Inception (GoogLeNet)</li>

            <li>MobileNet</li>

            <li>VGG (Visual Geometry Group)</li>

            <li>AlexNet</li>

        </ul>

    </section>

    <section>

        <h2>Параметри та кількість операцій</h2>

        <p>Нижче наведена таблиця, яка вказує на кількість параметрів та кількість операцій для класифікації одного

            зображення для кожної з вище зазначених моделей:</p>

        <table>

            <tr>

                <th>Модель</th>

                <th>Кількість параметрів</th>

                <th>Кількість операцій (приблизно)</th>

            </tr>

            <tr>

                <td>ResNet</td>

                <td>23.5M</td>

                <td>4.09B</td>

            </tr>

            <tr>

                <td>Inception</td>

                <td>22.8M</td>

                <td>5.74B</td>

            </tr>

            <tr>

                <td>MobileNet</td>

                <td>4.2M</td>

                <td>569M</td>

            </tr>

            <tr>

                <td>VGG</td>

                <td>138M</td>

                <td>15.5B</td>

            </tr>

            <tr>

                <td>AlexNet</td>

                <td>61M</td>

                <td>725M</td>

            </tr>

        </table>

    </section>

    <section>

        <h2>Додати нову модель</h2>

        <form id="add-model-form">

            <label for="model-name">Назва моделі:</label>

            <input type="text" id="model-name" name="model-name" required>

            <label for="model-parameters">Кількість параметрів:</label>

            <input type="text" id="model-parameters" name="model-parameters" required>

            <label for="model-operations">Кількість операцій (приблизно):</label>

            <input type="text" id="model-operations" name="model-operations" required>

            <button type="submit">Додати модель</button>

        </form>

    </section>

    <footer>

        <p>&copy; 2023 Класифікація нейронних мереж</p>

    </footer>

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

</body>

</html>

**Текст CSS файлу сайту**

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f0f0f0;

}

header {

    background-color: #333;

    color: #fff;

    text-align: center;

    padding: 20px;

}

h1 {

    margin: 0;

}

section {

    margin: 20px;

    background-color: #fff;

    padding: 20px;

    border: 1px solid #ddd;

    border-radius: 5px;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

h2 {

    font-size: 24px;

    margin-top: 0;

}

ul.no-bullets {

    list-style-type: none;

    padding-left: 0;

}

ul li {

    margin-bottom: 10px;

}

table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 10px;

}

th, td {

    border: 1px solid #ddd;

    padding: 8px;

    text-align: left;

}

th {

    background-color: #f2f2f2;

}

footer {

    text-align: center;

    padding: 10px;

    background-color: #333;

    color: #fff;

}

.no-bullets {

    list-style-type: none;

    padding-left: 20px; /\* Додайте відступ зліва від краю сторінки \*/

}

.no-bullets li {

    text-align: center;

}

.no-bullets li img {

    display: block;

    margin: 0 auto;

}

/\* Центрування контенту \*/

.center {

    text-align: center;

}

/\* Стилізація кнопок "Наступний" і "Попередній" \*/

.nav-button {

    background-color: transparent;

    border: none;

    font-size: 24px;

    cursor: pointer;

}

/\* Стилізація контейнера для кнопок \*/

.nav-button-container {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin-top: 20px; /\* Встановіть потрібний відступ від інформації про датасет \*/

}

.dataset-tile {

    background-color: #fff;

    padding: 20px;

    border: 1px solid #ddd;

    border-radius: 5px;

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

    text-align: center;

}

/\* Стилізація зображення внутрішньо блоку \*/

.dataset-tile img {

    display: block;

    margin: 0 auto;

    width: 150px; /\* Задайте потрібну ширину \*/

    height: auto; /\* Якщо необхідно підтримувати співвідношення сторін \*/

}

/\* Стилізація заголовка та опису \*/

.dataset-tile h3 {

    font-size: 24px;

    margin-top: 0;

}

section h2 {

    font-size: 24px;

    color: #333;

}

#add-model-form {

    background-color: #f4f4f4;

    border: 1px solid #ccc;

    padding: 20px;

    border-radius: 5px;

    margin-top: 10px;

}

#add-model-form label {

    font-weight: bold;

}

#add-model-form input[type="text"] {

    width: 100%;

    padding: 10px;

    margin-bottom: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

#add-model-form button {

    background-color: #333;

    color: #fff;

    border: none;

    padding: 10px 20px;

    border-radius: 5px;

    cursor: pointer;

}

#add-model-form button:hover {

    background-color: #555;

}

**Текст JavaScript файлу сайту**

var datasets = [

    {

       name: "ImageNet",

       description: "Великий набір з мільйонами зображень та 1000 категорій для класифікації.",

       link: "https://www.image-net.org/",

       imageUrl: "src/images/imagenet-logo.jpg"

   },

   {

       name: "COCO (Common Objects in Context)",

       description: "Набір даних для сегментації та класифікації об'єктів в контексті.",

       link: "https://cocodataset.org/",

       imageUrl: "src/images/coco-logo.png"

   },

   {

       name: "MNIST",

       description: "Набір рукописних цифр для класифікації.",

       link: "http://yann.lecun.com/exdb/mnist/",

       imageUrl: "src/images/mnist-logo.png"

   },

   {

       name: "CIFAR-10",

       description: "Набір даних із 10 класів об'єктів для класифікації.",

       link: "https://www.cs.toronto.edu/~kriz/cifar.html",

       imageUrl: "src/images/cifar-logo.jpg"

   }

];

var currentDatasetIndex = 0;

var datasetContainer = document.getElementById('dataset-container');

var prevButton = document.getElementById('prev-btn');

var nextButton = document.getElementById('next-btn');

function showDataset(index) {

    var dataset = datasets[index];

    var datasetHtml = `

        <div class="dataset-tile">

            <img src="${dataset.imageUrl}" alt="${dataset.name} логотип" width="150">

            <h3>  <a href="${dataset.link}">${dataset.name}</a></h3>

            <p>${dataset.description}</p>

        </div>

    `;

    datasetContainer.innerHTML = datasetHtml;

}

// Показати перший датасет при завантаженні сторінки

showDataset(currentDatasetIndex);

// Обробники подій для кнопок "Попередній датасет" і "Наступний датасет"

prevButton.addEventListener('click', function() {

    currentDatasetIndex = (currentDatasetIndex - 1 + datasets.length) % datasets.length;

    showDataset(currentDatasetIndex);

});

nextButton.addEventListener('click', function() {

    currentDatasetIndex = (currentDatasetIndex + 1) % datasets.length;

    showDataset(currentDatasetIndex);

});

const addModelForm = document.getElementById("add-model-form");

const modelContainer = document.getElementById("dataset-container");

addModelForm.addEventListener("submit", function (e) {

    e.preventDefault();

    const modelName = addModelForm.elements["model-name"].value;

    const modelParameters = addModelForm.elements["model-parameters"].value;

    const modelOperations = addModelForm.elements["model-operations"].value;

    const newRow = document.createElement("tr");

    newRow.innerHTML = `

        <td>${modelName}</td>

        <td>${modelParameters}</td>

        <td>${modelOperations}</td>

    `;

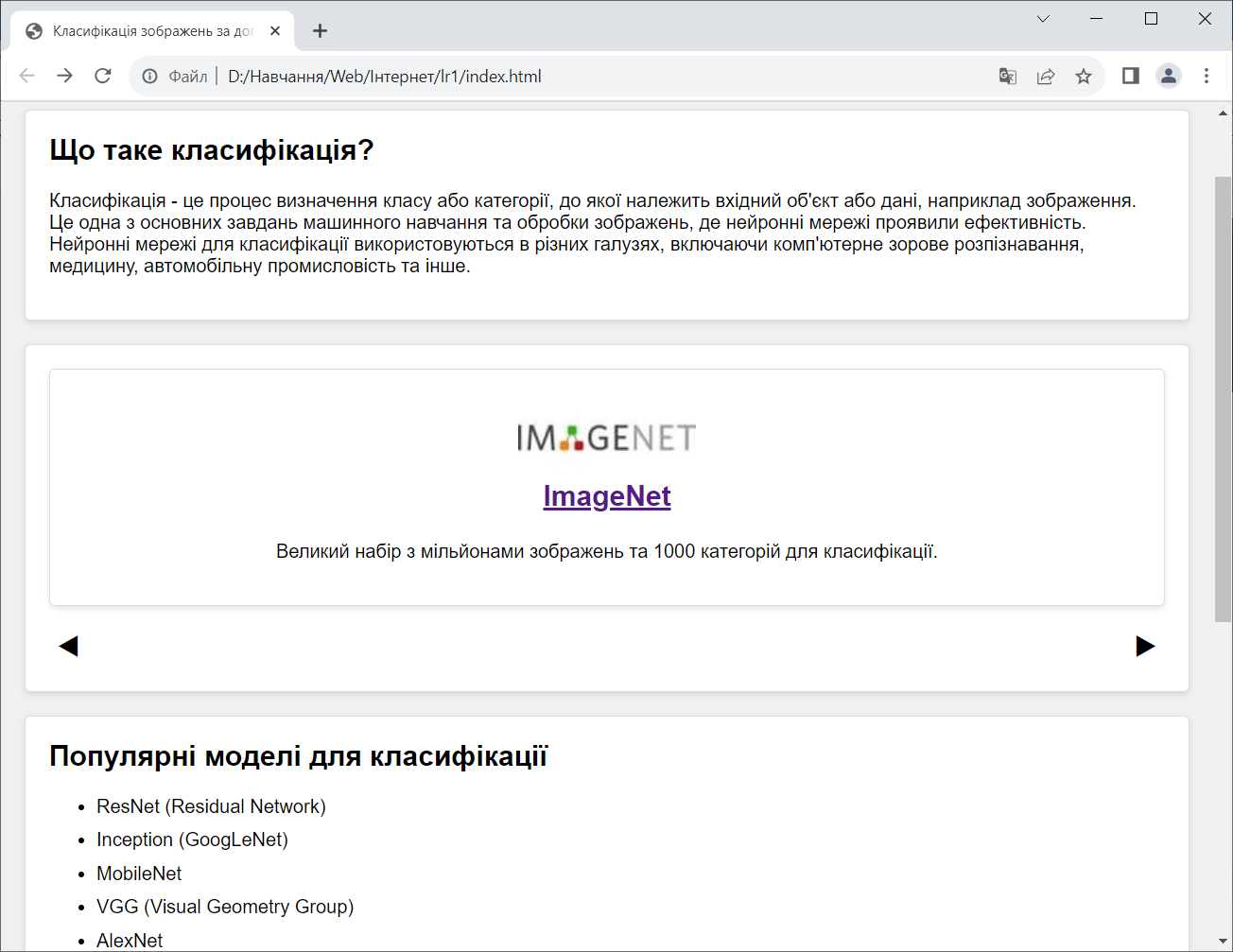
    const table = document.querySelector("table");

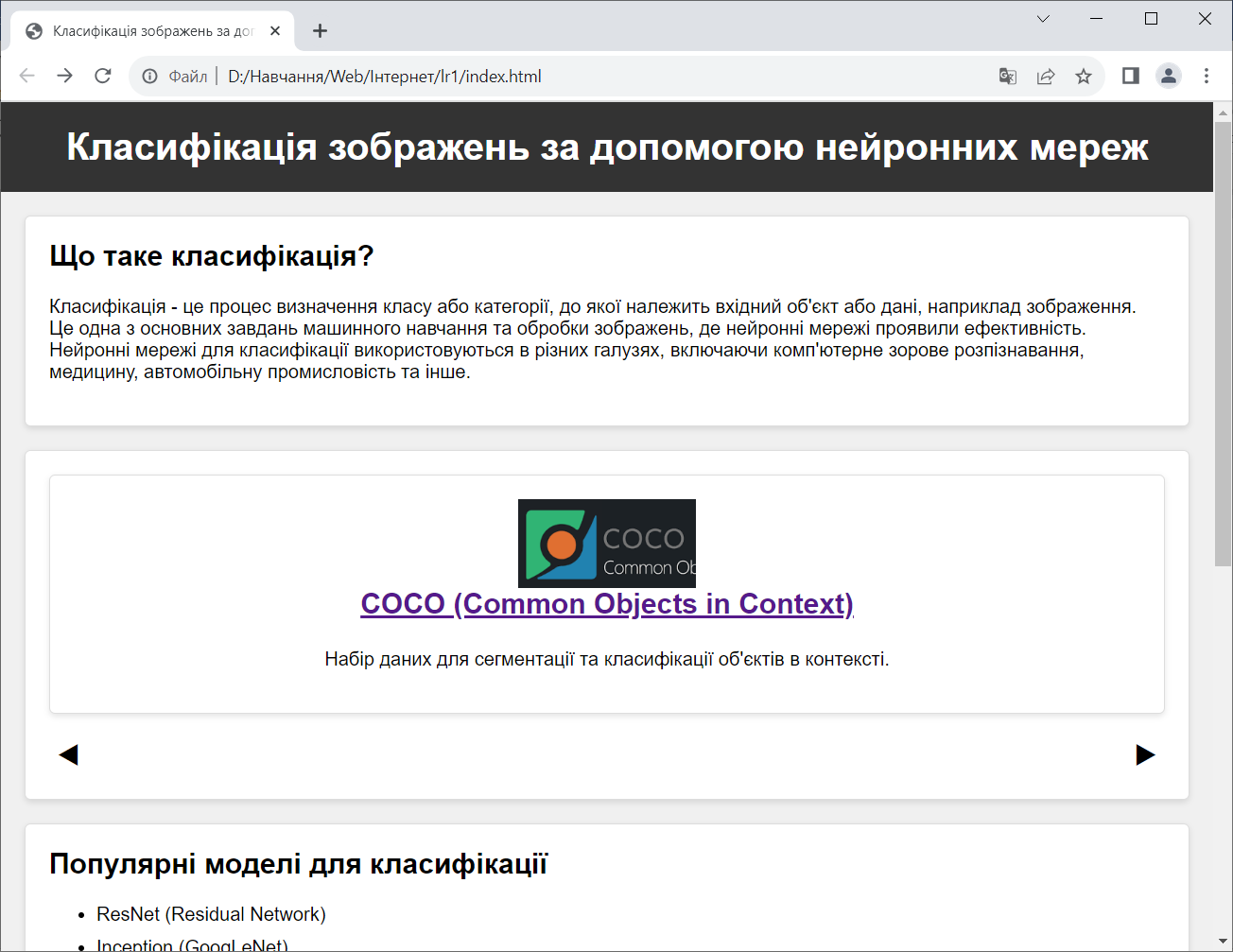
    table.appendChild(newRow);

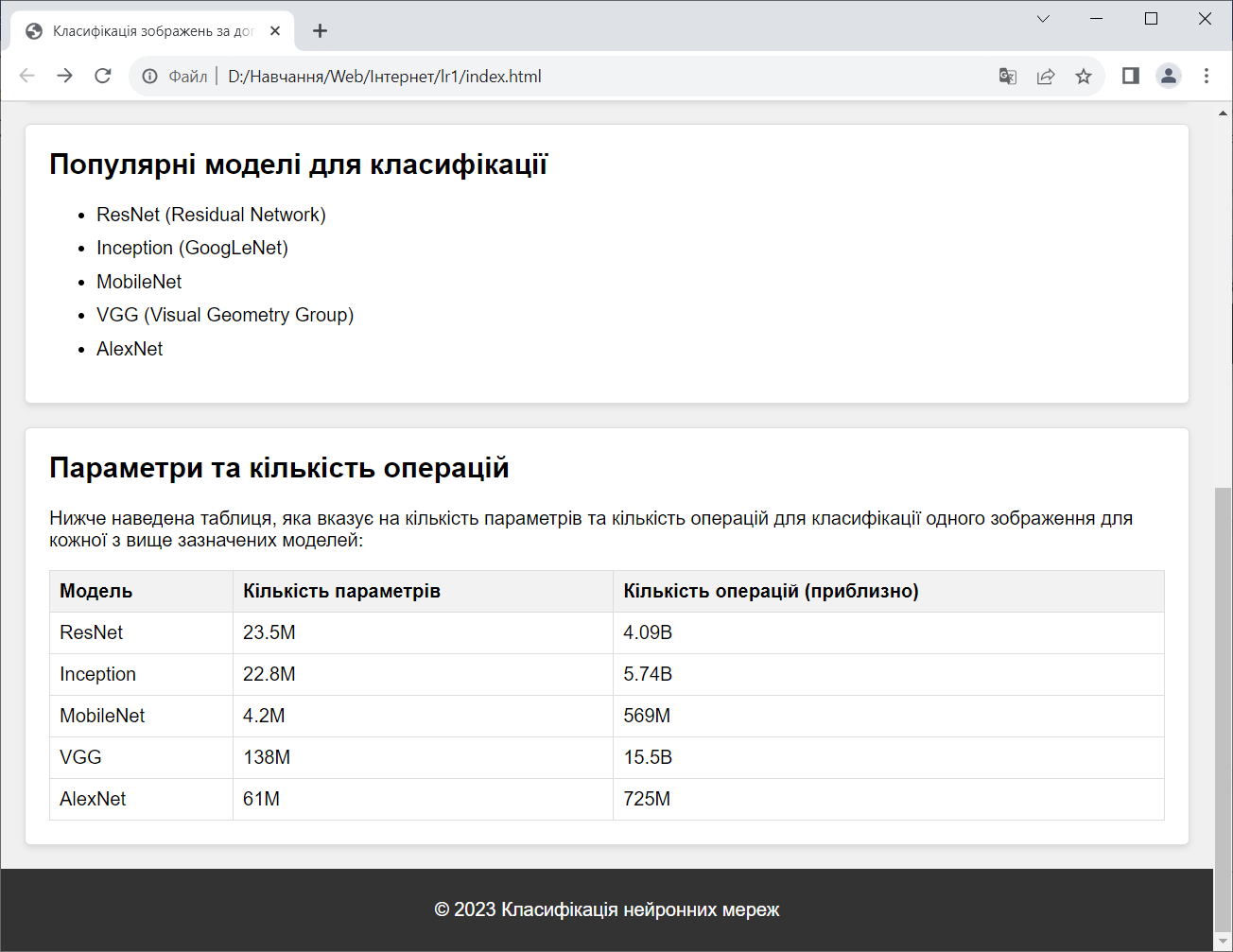
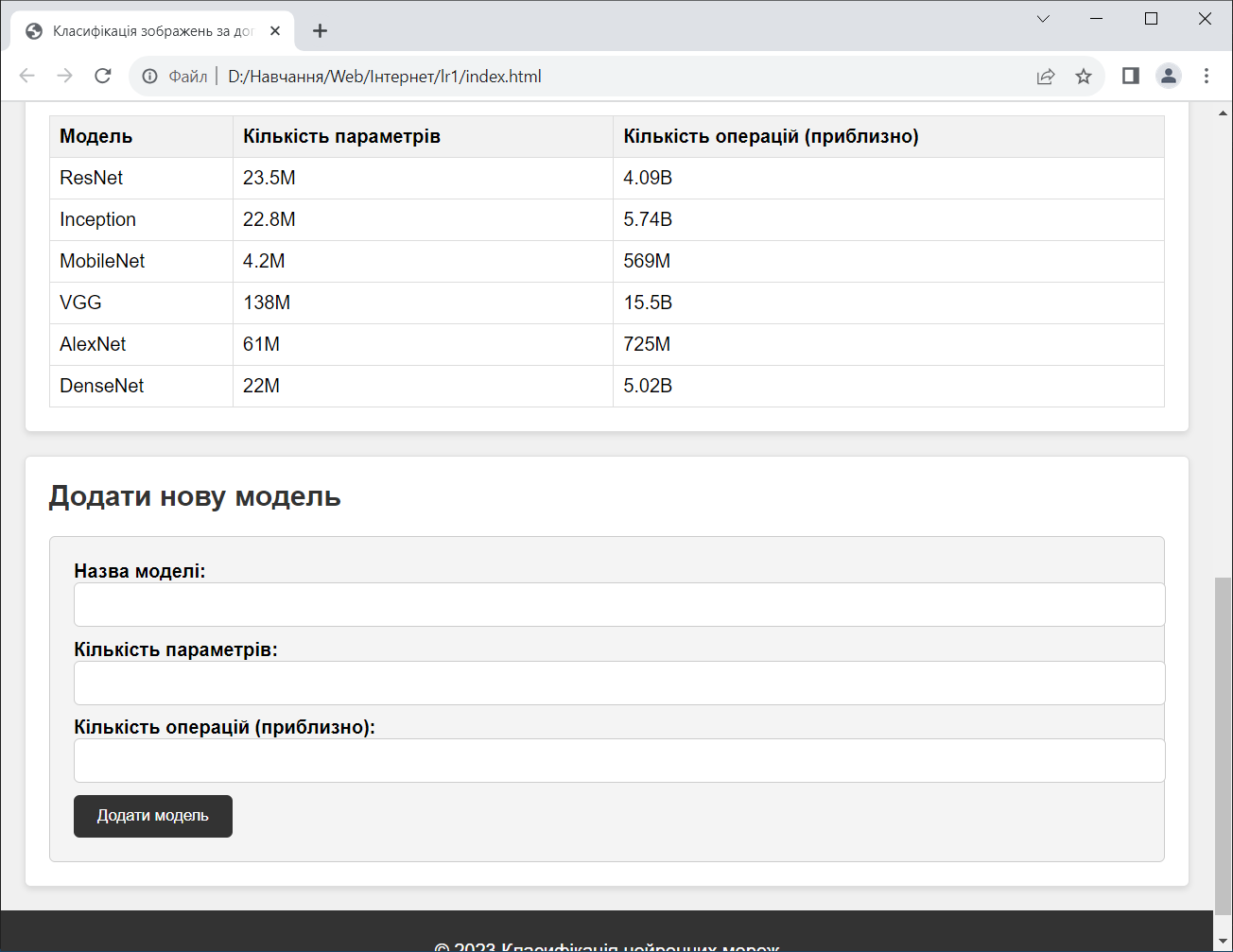
    addModelForm.reset();

});

**Демонстрація сайту**





**Висновки**

JavaScript (JS) - це мова програмування, яка використовується у веб-розробці для створення інтерактивних веб-сайтів і додатків. Вона дозволяє створювати динамічний контент на веб-сторінках, маніпулювати елементами сторінки, визначати обробники подій (наприклад, клік миші або натискання клавіш), валідувати дані у веб-формах і створювати анімації та візуалізації.

JavaScript допомагає покращити взаємодію користувачів з веб-сайтом та додає динаміку до статичних веб-сторінок. Ця мова також дозволяє використовувати сторонні бібліотеки і фреймворки для розробки більш складних і функціональних веб-застосунків.

Узагальнюючи, JavaScript є важливим інструментом у веб-розробці, який допомагає створювати більш інтерактивні та ефективні веб-сайти та додатки для користувачів.