

Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування

Навчально-науковий інститут кібернетики, інформаційних  
технологій та інженерії  
Кафедра обчислювальної техніки

**04-04-245М**

## **МЕТОДИЧНІ ВКАЗІВКИ**

до лабораторних робіт з навчальної дисципліни  
**«Web-програмування» (частина 1)**  
для здобувачів вищої освіти  
першого (бакалаврського) рівня  
за освітньо-професійною програмою  
«Комп'ютерна інженерія»  
спеціальності 123 «Комп'ютерна інженерія»  
та освітньо-професійною програмою  
«Інформаційна безпека»  
спеціальності 125 «Кібербезпека та захист інформації»  
денної та заочної форми навчання

Рекомендовано  
науково-методичною радою  
з якості ННІКІТІ  
Протокол № 4 від 24.02.2025 р.

Рівне – 2025

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Web-програмування» (частина 1) для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» та освітньо-професійною програмою «Інформаційна безпека» спеціальності 125 «Кібербезпека та захист інформації» денної та заочної форми навчання. [Електронне видання] / Бойчура М. В., Сидор А. І. – Рівне : НУВГП, 2025. – 38 с.

Укладачі: Бойчура М. В., к.т.н., доцент кафедри  
обчислювальної техніки;  
Сидор А. І., к.т.н., в.о. завідувача кафедри  
обчислювальної техніки.

Відповідальний за випуск: Сидор А. І., к.т.н., в.о. завідувача  
кафедри обчислювальної техніки.

Керівник групи забезпечення  
спеціальності  
123 «Комп'ютерна інженерія» Сидор А. І.

Керівник групи забезпечення  
спеціальності  
125 «Кібербезпека та захист інформації» Назарук В. Д.

© М. В. Бойчура,  
А. І. Сидор, 2025  
© НУВГП, 2025

## ЗМІСТ

Вступ .....	4
Лабораторна робота №1. Розмітка Web-сторінки засобами HTML. Теги. Атрибути .....	6
Лабораторна робота №2. Структурування Web-сайту .....	21
Лабораторна робота №3. Вступ до CSS .....	25
Лабораторна робота №4. Основи JavaScript .....	31
Рекомендована література .....	37

## Вступ

Розвиток Web-технологій суттєво змінив підхід до зберігання, обробки та представлення інформації. Сьогодні Web-програмування є однією з найважливіших дисциплін у сфері ІТ. Воно охоплює широкий спектр технологій, інструментів та мов програмування, що використовуються для створення Web-сайтів, Web-застосунків та інтерактивних сервісів.

Навчальна дисципліна «Web-програмування» забезпечує фундаментальні знання та практичні навички, необхідні для розробки сучасних Web-ресурсів. Вона сприяє розумінню основних принципів створення Web-сторінок, їхньої структуризації, стилізації, динамічного керування контентом та взаємодії з користувачем. Вивчення дисципліни дозволяє студентам оволодіти навичками роботи з HTML, CSS, JavaScript та іншими супутніми технологіями.

HTML (HyperText Markup Language) є основою будь-якого Web-документа. Ця мова розмітки використовується для визначення структури Web-сторінки за допомогою тегів та атрибутів. Вивчення HTML дає змогу студентам створювати статичні Web-сторінки, додавати мультимедійні елементи та посилання, а також створювати форми для збору даних.

CSS (Cascading Style Sheets) відповідає за стилізацію Web-сторінок, надаючи їм візуальну привабливість та адаптивність. Використання CSS дозволяє змінювати кольори, шрифти, розміщення елементів на Web-сторінці, а також створювати складні анімації та ефекти. Завдяки CSS студенти зможуть адаптувати Web-сторінки до різних пристроїв та розширень екрану.

JavaScript є мовою програмування, яка забезпечує динамічну взаємодію користувача з Web-сторінкою. Вона

дозволяє змінювати вміст сторінки без її перезавантаження, обробляти події, здійснювати перевірку введених даних у формах та взаємодіяти з сервером за допомогою AJAX. Розуміння основ JavaScript дозволяє створювати інтерактивні Web-застосунки та покращувати користувацький досвід.

Частина 1 дисципліни «Web-програмування» включає лабораторні роботи, спрямовані на закріплення теоретичних знань шляхом виконання заданого переліку завдань. Вони дозволяють студентам:

- навчитися створювати та стилізувати Web-сторінки;
- структурувати Web-сайти;
- додавати інтерактивні елементи;
- оптимізувати продуктивність Web-застосунків;
- навчитися користуватись фреймворком Bootstrap.

Завдяки лабораторним роботам студенти отримують можливість реалізовувати власні Web-проекти, починаючи від простих Web-сторінок, поступово ускладнюючи та розширюючи їх аж до розмірів Web-сайту з деякими інтегрованими інтерактивними елементами; у частині 2 даної дисципліни передбачається подальша розробка із використанням JavaScript-фреймворків. Практичний підхід сприяє формуванню професійних навичок, необхідних для подальшого розвитку у напрямку Web-програмування.

# **Лабораторна робота №1**

## **Розмітка Web-сторінки засобами HTML. Теги.**

### **Атрибути**

#### **Мета**

- Ознайомитись із синтаксисом мови HTML, зокрема тегами та атрибутами.
- Набути навички створення Web-сторінок.
- Навчитись формувати Web-сторінки за допомогою мовних моделей штучного інтелекту.

#### **Завдання**

1. Засобами мовних моделей штучного інтелекту сформувати Web-сторінку, яка міститиме не менше 100 різноманітних тегів мови HTML (без врахування закриваючих тегів).

2. Засобами мовних моделей штучного інтелекту додати до Web-сторінки понад 70 різноманітних атрибутів для стилізації.

3. Виконати верифікацію HTML-документу за допомогою спеціалізованих інструментів.

#### **Варіанти виконання завдань**

1. Автобіографія. Створіть сторінку з описом Вашого життя, хобі, досягнень. Додайте фотографії, таблиці з успішністю, списки інтересів.

2. Інтернет-магазин. Розробіть сторінку з описом товарів, їх характеристиками, цінами та способами доставки.

3. Сторінка кафедри. Опишіть викладачів, дисципліни, які вони викладають, додайте фотографії та таблиці з розкладом.

4. Аптека. Створіть сторінку з переліком ліків, їх кількістю, цінами та описом.

5. Футбольний комітет. Опишіть керівний склад, список команд, додайте фотографії та таблиці з результатами.

6. Громадська організація. Створіть сторінку з описом учасників, напрямків діяльності та відповідальних осіб.

7. Студентський гурток. Опишіть учасників гуртка, розклад заходів, додайте фотографії.

8. Косметична компанія. Розробіть сторінку з описом продукції, характеристиками та цінами.

9. Політична організація. Опишіть керівний склад, напрямки діяльності, додайте фотографії.

10. Олімп. Створіть сторінку з описом грецьких або римських богів та їх сфер «відповідальності».

11. Інформаційний відділ. Опишіть персональний склад відділу, перелік послуг, додайте фотографії.

12. Студентська група. Створіть сторінку з переліком студентів та їх персональними даними.

13. Банк. Опишіть базу даних клієнтів, наявні послуги, додайте фотографії.

14. Кінотеатр. Розробіть сторінку з переліком фільмів, прайс-листом, додайте фотографії.

15. Асоціація кінолюбителів. Опишіть членів асоціації, переможців кінопремій, додайте анонси фільмів.

16. Сайт наукових новин. Створіть сторінку з поточними новинами та складом команди журналістів.

17. Агентство нерухомості. Опишіть пропозиції щодо продажу та купівлі житла і додайте фотографії.

18. Автомобільний салон. Розробіть сторінку з переліком автомобілів, їх характеристиками та цінами.

19. Асоціація книголюбів. Опишіть осередки асоціації, рекомендовані книги, додайте фотографії.

20. Студентська рада. Створіть сторінку зі складом ради, проєктами рішень, додайте фотографії.

21. Музей. Опишіть експонати, розклад екскурсій, додайте фотографії.

22. Ресторан. Розробіть сторінку з меню, описом страв, додайте фотографії.

23. Фітнес-клуб. Опишіть послуги, розклад занять, додайте фотографії.

24. Туристичне агентство. Створіть сторінку з описом турів, цінами, додайте фотографії.

25. Школа. Опишіть вчителів, дисципліни, додайте фотографії.

26. Бібліотека. Розробіть сторінку з переліком книг, їх описом, додайте фотографії.

27. Театр. Опишіть вистави, акторів, додайте фотографії.

28. Музична школа. Створіть сторінку з описом курсів, викладачів, додайте фотографії.

29. Фермерське господарство. Опишіть продукцію, способи доставки, додайте фотографії.

30. ІТ-компанія. Розробіть сторінку з описом послуг, командою, додайте фотографії.

31. Медіа-агентство. Опишіть послуги, команду, додайте фотографії.

32. Спортивний клуб. Створіть сторінку з описом тренерів, розкладом занять, додайте фотографії.

33. Фотостудія. Опишіть послуги, портфоліо, додайте фотографії.

34. Юридична фірма. Розробіть сторінку з описом послуг, командою, додайте фотографії.

35. Ветеринарна клініка. Опишіть послуги, лікарів та додайте фотографії.

36. Садовий центр. Створіть сторінку з описом рослин, цінами, додайте фотографії.

37. Арт-галерея. Опишіть виставки, художників, додайте фотографії.

38. Маркетингова агенція. Розробіть сторінку з описом послуг, кейсами, додайте фотографії.



39. Мережа кафе. Опишіть меню, філії, додайте фотографії.

40. Онлайн-курси. Створіть сторінку з описом курсів, викладачів, додайте фотографії.

### Вимоги до розв'язків завдань

- Теги. Використовуйте різноманітні теги: заголовки (<h1>-<h6>), абзаци (<p>), списки (<ul>, <ol>), таблиці (<table>), зображення (<img>), посилання (<a>), форми (<form>), кнопки (<button>) тощо (табл. 1).

- Атрибути. Застосовуйте атрибути для стилізації: class, id, style, src, alt, href, width, height, target, title, placeholder тощо (табл. 2).

- Семантика. Використовуйте семантичні теги: <header>, <nav>, <main>, <section>, <article>, <aside>, <footer> (табл. 1).

- Верифікація. Виконайте верифікацію HTML-документу за допомогою W3C Validator.

- Цифрова доступність. Забезпечте цифрову доступність сторінки за допомогою атрибутів aria-\* та семантичних тегів.

Таблиця 1

#### Перелік тегів мови HTML5

№ п/п	Назва тега	Опис
<i>Структура документу</i>		
1.	<!DOCTYPE>	оголошення типу документа (визначає версію HTML)
2.	<html>	основний контейнер HTML-документа
3.	<head>	містить мета-інформацію документа
4.	<title>	заголовок Web-сторінки (відображається на вкладці браузера)
5.	<body>	основний контент Web-сторінки

продовження табл. 1

<i>Заголовки, текст і форматування</i>		
6.	<h1>	заголовки різних рівнів (від найбільшого <h1> до найменшого <h6>)
7.	<h2>	
8.	<h3>	
9.	<h4>	
10.	<h5>	
11.	<h6>	
12.	<p>	абзац тексту
13.	 	перенесення рядка
14.	<hr>	горизонтальна лінія
15.	<strong>	важливий текст (жирний)
16.	<b>	жирний текст (без семантичного значення)
17.	<em>	наголошений текст (курсив)
18.	<i>	курсивний текст (без семантичного значення)
19.	<small>	менший текст
20.	<mark>	виділений текст (жовтий фон за замовчуванням)
21.	<del>	перекреслений текст (видалений)
22.	<ins>	підкреслений текст (вставлений)
23.	<sub>	нижній індекс
24.	<sup>	верхній індекс
25.	<abbr>	скорочення (з поясненням при наведенні)
26.	<cite>	назва твору (курсив)
27.	<code>	код програми (моноширинний шрифт)
28.	<pre>	попередньо відформатований текст
29.	<s>	перекреслений текст (застарілий або некоректний)
30.	<u>	підкреслений текст
31.	<q>	коротка цитата
32.	<blockquote>	довга цитата (із відступом)
33.	<dfn>	виділення визначення терміну
34.	<var>	позначає змінну в коді

продовження табл. 1

35.	<kbd>	вказує текст, що користувач повинен ввести (наприклад, клавіатурні комбінації)
36.	<samp>	виведення комп'ютерного коду
37.	<bdi>	вказує незалежний напрямок тексту
38.	<bdo>	визначає напрямок тексту (LTR або RTL)
39.	<wbr>	дозволяє браузеру самостійно визначити місце для перенесення слова
<i>Списки</i>		
40.	<ul>	невпорядкований (маркований) список
41.	<ol>	впорядкований (нумерований) список
42.	<li>	елемент списку
43.	<dl>	список визначень
44.	<dt>	термін у списку визначень
45.	<dd>	опис терміну в списку визначень
<i>Посилання та навігація</i>		
46.	<a>	гіперпосилання
47.	<nav>	навігаційний розділ Web-сайту
48.	<area>	активна область зображення-карти
49.	<map>	визначає карту зображень
<i>Медіаконтент (зображення, відео, аудіо)</i>		
50.	<img>	відображає зображення
51.	<figure>	контейнер для зображення та підпису
52.	<figcaption>	підпис до <figure>
53.	<audio>	додає аудіофайл
54.	<video>	додає відеофайл
55.	<source>	визначає джерело мультимедіа (для <video> або <audio>)
56.	<track>	додає субтитри до відео
<i>Форми та введення даних</i>		
57.	<form>	контейнер для форми
58.	<input>	поле введення
59.	<label>	підпис для елементів форми
60.	<textarea>	поле введення багаторядкового тексту
61.	<select>	випадаючий список

продовження табл. 1

62.	<option>	опція у випадяючому списку
63.	<optgroup>	група опцій у випадяючому списку
64.	<button>	кнопка
65.	<fieldset>	групує елементи форми
66.	<legend>	заголовок для <fieldset>
67.	<datalist>	список доступних варіантів для <input>
68.	<output>	відображає результат обчислень
<i>Таблиці</i>		
69.	<table>	таблиця
70.	<caption>	назва таблиці
71.	<thead>	заголовок таблиці
72.	<tbody>	тіло таблиці
73.	<tfoot>	підвал таблиці
74.	<tr>	рядок таблиці
75.	<th>	заголовок колонки таблиці
76.	<td>	комірка таблиці
77.	<colgroup>	групування колонок
78.	<col>	визначає властивості колонок
<i>Семантичні та структурні теги</i>		
79.	<header>	заголовок секції або Web-сторінки
80.	<footer>	нижній колонтитул Web-сторінки або секції
81.	<section>	секція контенту
82.	<article>	незалежний контент (стаття, блог)
83.	<aside>	боковий контент (сайдбар)
84.	<main>	основний контент Web-сторінки
85.	<summary>	заголовок для <details>
86.	<details>	блок інформації, який розгортається при кліку
87.	<mark>	виділений текст
88.	<time>	дата/час у машиночитному форматі
89.	<address>	контактна інформація
<i>Скрипти, стилі та мета-дані</i>		
90.	<script>	вставка JavaScript-коду

продовження табл. 1

91.	<noscript>	вміст, що відображається, якщо браузер не підтримує JavaScript
92.	<style>	вбудовані CSS-стилі
93.	<meta>	мета-інформація про Web-сторінку
94.	<link>	підключення зовнішніх ресурсів (CSS, іконки)
95.	<base>	визначає базовий URL для відносних посилань
<i>Менш відомі та спеціалізовані теги</i>		
96.	<ruby>	анотація символів (наприклад, японських ієрогліфів)
97.	<rt>	текстова анотація для <ruby>
98.	<rp>	відображення анотацій у браузерах, що не підтримують <ruby>
99.	<template>	шаблон контенту, який не відображається одразу
100.	<slot>	вставка контенту у Web Components

Таблиця 2

Перелік атрибутів мови HTML

№ п/п	Назва атрибуту	Призначення атрибуту
<i>Глобальні атрибути (працюють майже з усіма тегами)</i>		
1.	accesskey	визначає гарячі клавіші для швидкого доступу до елемента
2.	autocapitalize	контролює автоматичну зміну регістру введеного тексту (наприклад, що перша літера – велика)
3.	autofocus	автоматично встановлює фокус на елемент після завантаження сторінки
4.	class	присвоює елементу один або декілька класів CSS

продовження табл. 2

5.	<code>contenteditable</code>	дозволяє робити вміст елемента редагованим
6.	<code>dir</code>	встановлює напрямок тексту (“ltr” – зліва направо, “rtl” – справа наліво)
7.	<code>draggable</code>	визначає, чи можна перетягувати елемент (“true” або “false”)
8.	<code>enterkeyhint</code>	підказує, який тип клавіші “Enter” повинен використовуватися (наприклад, “search”, “done”)
9.	<code>hidden</code>	ховає елемент, роблячи його невидимим для користувача
10.	<code>id</code>	унікальний ідентифікатор елемента на сторінці
11.	<code>inert</code>	робить елемент неактивним
12.	<code>inputmode</code>	визначає режим введення для мобільних клавіатур (наприклад, “numeric” для цифр)
13.	<code>is</code>	використовується для створення користувацьких елементів (Web Components)
14.	<code>itemid</code>	використовуються в мікроданих (microdata) для семантичного опису Web-сторінки
15.	<code>itemprop</code>	
16.	<code>itemref</code>	
17.	<code>itemscope</code>	
18.	<code>itemtype</code>	
19.	<code>lang</code>	визначає мову вмісту елемента
20.	<code>nonce</code>	використовується для підвищення безпеки, дозволяючи завантаження довіреного коду
21.	<code>part</code>	дозволяє стилізувати Shadow DOM через CSS
22.	<code>popover</code>	використовується для створення модальних вікон

продовження табл. 2

23.	role	визначає роль елемента для засобів доступності
24.	slot	використовується в Web Components для організації вмісту
25.	spellcheck	вказує, чи потрібно перевіряти орфографію введеного тексту
26.	style	додає вбудовані CSS-стилі до елемента
27.	tabindex	визначає порядок переходу між елементами при натисканні клавіші Tab
28.	title	додає текстову підказку, яка з'являється при наведенні курсора
29.	translate	вказує, чи потрібно перекладати вміст елемента
<i>Атрибути форм та введення даних</i>		
30.	accept	визначає допустимі типи файлів для завантаження через <code>&lt;input type="file"&gt;</code>
31.	autocomplete	вказує, чи дозволене браузеру автозаповнення форми
32.	capture	визначає, чи використовувати камеру або мікрофон за замовчуванням при завантаженні файлу
33.	checked	встановлює прапорець або радіокнопку у вибраній стан
34.	dirname	вказує напрямок введеного тексту (ltr або rtl)
35.	disabled	блокує взаємодію користувача з елементом
36.	form	прив'язує елемент до конкретної форми

продовження табл. 2

37.	formaction	вказує URL, куди буде відправлена форма (для <code>&lt;input type="submit"&gt;</code> )
38.	formenctype	визначає тип кодування даних при відправленні форми
39.	formmethod	визначає HTTP-метод для відправлення форми (GET, POST)
40.	formnovalidate	вимикає перевірку коректності введених даних
41.	formtarget	визначає, де відкривати результат відправлення форми ( <code>_blank</code> , <code>_self</code> )
42.	height	задає висоту елемента
43.	list	зв'язує поле введення з випадającym списком <code>&lt;datalist&gt;</code>
44.	max	встановлює максимальне значення для числового введення
45.	maxlength	обмежує кількість символів у полі введення
46.	min	встановлює мінімальне значення для числового введення
47.	multiple	дозволяє вибір декількох файлів або значень
48.	name	визначає ім'я елемента (важливо для передачі даних на сервер)
49.	pattern	додає регулярний вираз для перевірки введених даних
50.	placeholder	відображає текст-підказку всередині поля введення
51.	readonly	робить поле введення доступним лише для читання
52.	required	змушує користувача заповнити поле перед відправленням форми
53.	size	визначає ширину поля введення



продовження табл. 2

54.	step	визначає крок для числового введення
55.	value	встановлює значення за замовчуванням
56.	width	задає ширину елемента
<i>Атрибути мультимедіа</i>		
57.	autoplay	автоматично запускає аудіо чи відео після завантаження сторінки
58.	controls	додає елементи керування для відтворення медіа
59.	crossorigin	визначає, чи дозволено завантажувати медіа з іншого домену
60.	loop	повторює відтворення відео або аудіо
61.	muted	вимикає звук для відео або аудіо за замовчуванням
62.	poster	встановлює зображення-заставку для відео
63.	preload	визначає, чи завантажувати медіа перед відтворенням
64.	src	вказує шлях до медіафайлу
<i>Атрибути для роботи з посиланнями та навігацією</i>		
65.	download	дозволяє завантажити файл при натисканні на посилання
66.	href	визначає URL для посилання
67.	hreflang	вказує мову сторінки, на яку веде посилання
68.	ping	відправляє запити до вказаних серверів при переході за посиланням
69.	referrerpolicy	контролює, які заголовки передавати при переході за посиланням

продовження табл. 2

70.	rel	визначає зв'язок між поточною сторінкою та цільовою
71.	target	визначає, як відкривати посилання ( <code>_blank</code> , <code>_self</code> )
<i>Атрибути для таблиць</i>		
72.	colspan	визначає, скільки стовпців займає комірка
73.	rowspan	визначає, скільки рядків займає комірка
74.	headers	пов'язує комірку <code>&lt;td&gt;</code> із заголовком <code>&lt;th&gt;</code>

### Критерії оцінювання

1. Виконання вимог (40 балів):
  - використання не менше, ніж 100 тегів (10 балів);
  - застосування не менше, ніж 70 атрибутів для стилізації (10 балів);
  - успішна верифікація HTML-документу (10 балів);
  - коректне розміщення семантичних тегів (10 балів).
2. Якість коду (30 балів):
  - чистота та структурованість коду (10 балів);
  - відповідність стандартам W3C (10 балів);
  - цифрова доступність Web-сторінки (10 балів).
3. Креативність (20 балів):
  - оригінальність дизайну (10 балів);
  - використання додаткових елементів (10 балів).
4. Документація (10 балів):
  - наявність коментарів у кодї (5 балів);
  - опис функціональності Web-сторінки (5 балів).

## **Додаткова інформація**

Самостійна розробка Web-сторінки з понад 100 тегами мови HTML вимагає суттєвих витрат часу, що не завжди є обґрунтованим. Тому для досягнення поставленої цілі більш доречним є використання мовних моделей штучного інтелекту. З іншого боку, відповіді на запити у чатах найбільш популярних моделей ChatGPT та Gemini зачасту обмежуються декількома тисячами символами, чого є недостатньо. Таким чином варто у тих чи інших ситуаціях зважати на особливості, переваги та недоліки використовуваних моделей. Одним із хороших рішень є використання платформи Groq та доступних там моделей Llama різних версій. Також набуває популярності DeepSeek, що дозволяє також отримувати довгі відповіді.

Перевірку кількості згенерованих тегів також можна здійснювати засобами штучного інтелекту.

## **Контрольні запитання**

1. Що таке HTML?
2. Які основні теги HTML Ви знаєте?
3. Як створити заголовок у HTML?
4. Як додати зображення на сторінку?
5. Як створити посилання в HTML?
6. Як створити таблицю в HTML?
7. Як додати список у HTML?
8. Що таке семантичні теги?
9. Які атрибути використовуються для стилізації?
10. Що таке верифікація HTML?
11. Як додати кнопку в HTML?
12. Як створити фрейм в HTML?
13. Як додати аудіо на сторінку?
14. Як додати відео на сторінку?
15. Як створити меню навігації?
16. Як додати коментарі в HTML?

17. Як створити розділ з контентом?
18. Як додати футер на сторінку?
19. Як додати мета-теги в HTML?
20. Як створити випадаючий список?
21. Як додати фон на сторінку?
22. Як додати іконку на сторінку?
23. Як додати підказку до елемента?
24. Для чого використовується тег `<html>`?
25. Яке призначення тега `<head>`?
26. Яка роль тега `<body>`?
27. Що таке атрибут `alt`?
28. Що таке атрибут `target`?
29. Як вставити чекбокси на сторінку?
30. Як додати підпис до зображення?
31. Як зробити текст підкресленим?
32. Як зробити горизонтальну лінію?
33. Як реалізувати поле для введення паролю?
34. Що таке meta-теги?
35. Які є глобальні атрибути в HTML?
36. Як налаштувати гіперпосилання в нове вікно?
37. Як зробити перенесення рядка в HTML?
38. Як об'єднати комірки таблиці?
39. Які основні атрибути використовуються для таблиць?
40. Як додати альтернативний текст для зображення?

## **Лабораторна робота №2**

### **Структурування Web-сайту**

#### **Мета**

Навчитись структурувати Web-сторінки та Web-сайти за допомогою семантичних тегів HTML, фреймів та інших інструментів.

#### **Завдання**

1. Доповніть Web-сторінку, розроблену на лабораторній роботі №1, ще декількома сторінками, поєднавши їх пунктами навігаційного меню.
2. Використайте фрейми (<iframe>) для вбудовування контенту з інших джерел.
3. Організуйте коректну файловою структуру проекту.

#### **Вимоги до розв'язків завдань**

- Семантичні теги. Коректно використовуйте на всіх сторінках <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>. У якості прикладу структурування Web-сторінки можете використовувати рис. 1.
- Фрейми. Використовуйте <iframe> для вбудовування контенту.
- Файлова структура. Організуйте папку проекту, розмістивши там декілька файлів з роширенням \*.html та папку images з рисунками у ній.
- Навігація. Створіть меню з посиланнями на розділи Web-сайту.

#### **Критерії оцінювання**

1. Виконання вимог (40 балів):
  - використання семантичних тегів (10 балів);
  - застосування фреймів (10 балів);
  - організація файлової структури (10 балів);

- наявність навігації (10 балів).
2. Якість коду (30 балів):
    - чистота та структурованість коду (10 балів);
    - відповідність стандартам W3C (10 балів);
    - адаптивність Web-сторінки (10 балів).
  3. Креативність (20 балів):
    - оригінальність дизайну (10 балів);
    - використання додаткових елементів (10 балів).
  4. Документація (10 балів):
    - наявність коментарів у кодї (5 балів);
    - опис функціональності Web-сторінки (5 балів).

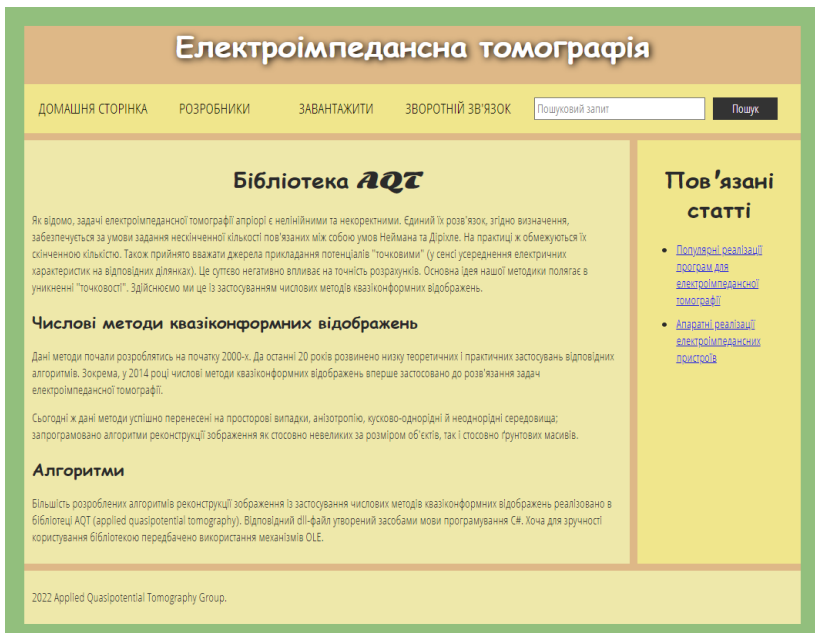


Рис. 1. Приклад коректного структурування Web-сайту

## Додаткова інформація

Структурування Web-сайту на початкових етапах вивчення Web-програмування є тривалим та трудомістким

процесом, який вимагає постійного консультування ментора або викладача. Ще одним ефективним інструментом є використання віртуальних асистентів, які забезпечуються, наприклад, компанією OpenAI (кнопка Explore GPTs). Можна й більш звичними підходами задавати запит до моделі із проханням сформувати Web-сайт на основі існуючої Web-сторінки. ChatGPT досить якісно справляється з цією задачею, хоча й має обмеження щодо довжини відповіді. Тому варто правильно формулювати запити. Загалом у цьому контексті доцільно звернути увагу на ресурс Chatbot Arena LLM Leaderboard на ресурсі HuggingFace.

### **Контрольні запитання**

1. Що таке структура Web-сайту?
2. Які основні елементи структури Web-сторінки?
3. Яке призначення тега <header>?
4. Яке призначення тега <nav>?
5. Як організувати навігаційне меню?
6. Що таке <aside> і як його використовувати?
7. Яке призначення <footer>?
8. Що таке семантична верстка?
9. Чому важливо використовувати семантичні теги?
10. Як створити багатосторінковий сайт?
11. Як зв'язати HTML-сторінки між собою?
12. Що таке файловий менеджмент у Web-розробці?
13. Як організувати коректну файлову структуру проекту?
14. У якій папці варто зберігати зображення та медіа-файли?
15. Які є методи вставки зовнішніх ресурсів на сторінку?
16. Що таке <iframe> і як його використовувати?
17. Як вставити YouTube-відео на сторінку?

18. Які є основні проблеми при структуризації Web-сайту?
19. Як створити бокову панель (sidebar)?
20. Як зробити фіксоване (fixed) меню?
21. Що таке таби (tabs) на сайті?
22. Як реалізувати вкладені меню (dropdown)?
23. Що таке hover-ефекти у навігації?
24. Як зробити активне підсвічування пунктів меню?
25. Як перевірити правильність структури сайту?
26. Які інструменти допомагають тестувати структуру сайту?
27. Як організувати внутрішнє посилання між сторінками сайту?
28. Як зробити шаблон сайту для подальшого використання?
29. У якій папці варто зберігати CSS-файли?
30. Як організувати навігацію між папками проєкту?
31. На яку Web-сторінку має вести посилання при натисненні на логотип сайту?
32. Який тег призначений для розміщення пунктів навігаційного меню?
33. Який тег зазвичай використовують у кнопках навігаційного меню для переходу між сторінками?
34. Яким має бути значення атрибуту target у пунктах навігаційного меню?
35. Як повинен називатись файл з домашньою сторінкою?
36. Чи може міститись footer всередині у <body>?
37. Яка виникне помилка, якщо допустити опіску в атрибуті src?
38. Як використовувати тег title на Web-сторінках?
39. Яка різниця між блочною та табличною версткою?
40. Для чого потрібний тег !DOCTYPE?



## Лабораторна робота №3

### Вступ до CSS

#### Мета

- Ознайомитись із каскадними таблицями стилів (CSS).
- Набути навички стилізації Web-сторінок за допомогою селекторів та властивостей.
- Ознайомитись із фреймворком Bootstrap.

#### Завдання

1. Створіть файл стилів style.css та підключіть його до усіх Web-сторінок, розроблених в межах попередніх лабораторних робіт.
2. Використовуйте різні типи селекторів: універсальні, класи, ідентифікатори, атрибути.
3. Застосуйте фреймворк Bootstrap для швидкої стилізації сторінки.

#### Вимоги до програмної реалізації

- Властивості. Застосовуйте властивості CSS для форматування тексту, кольорів, відступів, рамок.
- Фреймворки. Використовуйте Bootstrap для створення адаптивного дизайну.
- Анімації. Додайте анімації за допомогою CSS.
- Медіа-запити. Забезпечте адаптивність Web-сторінки за допомогою медіа-запитів.
- Селектори. Використовуйте різні типи селекторів: класів, ідентифікаторів, атрибутів, універсальні (табл. 3). Також варто звернути увагу й на наступні селектори:
  - атрибути: [атрибут = "значення"], [атрибут ~= "значення"], [атрибут |= "значення"], [атрибут ^= "значення"], [атрибут \$= "значення"], [атрибут \*= "значення"];

- псевдокласи: `:first-child`, `:last-child`, `:nth-child(число)`, `:nth-of-type(число)`, `:hover`, `:focus`, `:active`, `:checked`, `:visited`, `:link`;
- псевдоелементи: `::first-line`, `::first-letter`, `::selection`, `::before`, `::after`.

Таблиця 3

Перелік деяких селекторів та приклади їх використання

Тип селектора	Опис	Приклад	Пояснення
<i>Загальні селектори</i>			
Універсальний	вибирає всі елементи сторінці	<code>* { color: blue; }</code>	всі елементи стануть синіми
За назвою тега	вибирає всі елементи вказаного тега	<code>p { font-size: 16px; }</code>	усі абзаци матимуть розмір шрифту 16 пікселів
За класом	вибирає всі елементи з вказаним класом	<code>.red { color: red; }</code>	усі елементи з класом "red" стануть червоними
За ідентифікатором	вибирає елемент з вказаним ідентифікатором	<code>#header { background-color: yellow; }</code>	елемент з ідентифікатором "header" матиме жовтий фон
За атрибутом	вибирає всі елементи з вказаним атрибутом	<code>a[href] { text-decoration: none; }</code>	усі посилання не матимуть підкреслення
<i>Селектори-комбінатори</i>			

продовження табл. 3

Кома	вибирає всі елементи, розділені комою	h1, h2, h3 { font-weight: bold; }	усі заголовки h1, h2 та h3 будуть жирними
Нашадків	вибирає всі елементи, що є нащадками вказаного елемента	div p { color: green; }	усі абзаци всередині div будуть зеленими
Дочірній	вибирає всі елементи, що є безпосередніми дочірніми елементами вказаного елемента	ul > li { list-style: square; }	усі елементи li, що є безпосередніми дочірніми елементами ul, матимуть квадратні маркери списку
Усіх сусідніх елементів	вибирає всі елементи, що знаходяться на одному рівні з вказаним елементом та йдуть після нього	h2 ~ p { font-style: italic; }	усі параграфи, що йдуть після h2, будуть курсивними
Наступного сусіднього елемента	вибирає наступний елемент, що знаходиться на одному рівні з вказаним елементом	h2 + p { text-decoration: underline; }	наступний абзац після h2 буде підкресленим
<i>Селектори по псевдокласу та псевдоелементу</i>			

продовження табл. 3

По псевдокласу	вибирає елементи певному стані	<code>a:hover { color: orange; }</code>	посилання стануть помаранчевими при наведенні курсора
По псевдоелементу	вибирає частину елемента	<code>p::first-line { font-weight: bold; }</code>	перший рядок кожного абзацу буде потовщеним

### Критерії оцінювання

1. Виконання вимог (40 балів):
  - використання різних типів селекторів (10 балів);
  - застосування властивостей CSS (10 балів);
  - використання Bootstrap (10 балів);
  - адаптивність Web-сторінки (10 балів).
2. Якість коду (30 балів):
  - чистота та структурованість коду (10 балів);
  - відповідність стандартам W3C (10 балів);
  - наявність анімацій (10 балів).
3. Креативність (20 балів):
  - оригінальність дизайну (10 балів);
  - використання додаткових елементів (10 балів).
4. Документація (10 балів):
  - наявність коментарів у кодї (5 балів);
  - опис функціональності Web-сторінки (5 балів).

### Додаткова інформація

Стилізація Web-сторінок є дуже складною задачею на початкових етапах вивчення тем верстки Web-сторінок. Через що доцільно користуватись мовними моделями штучного інтелекту. Іншим ефективним інструментом,

який частково допомагає швидко та зручно стилізувати Web-сторінки є CSS-фреймворки. Одним із найбільш популярних є Bootstrap. Важливо, що Web-сторінки, сформовані за допомогою Bootstrap є більш зрозумілими, ніж у випадку використання звичайного CSS. Таким чином зручним варіантом стилізації Web-сторінок є використання мовних моделей штучного інтелекту, де у запиті просити виводити результат на основі фреймворку Bootstrap.

### **Контрольні запитання**

1. Що таке CSS?
2. Які типи селекторів Ви знаєте?
3. Як підключити CSS до HTML?
4. Як змінити колір тексту за допомогою CSS?
5. Як змінити фон сторінки за допомогою CSS?
6. Як змінити розмір шрифту за допомогою CSS?
7. Як змінити відступи за допомогою CSS?
8. Як змінити рамку елемента за допомогою CSS?
9. Як створити анімацію за допомогою CSS?
10. Як зробити сторінку адаптивною за допомогою CSS?
11. Як змінити позицію елемента за допомогою CSS?
12. Як змінити розмір елемента за допомогою CSS?
13. Як змінити вирівнювання тексту за допомогою CSS?
14. Як змінити колір фону елемента за допомогою CSS?
15. Як змінити колір рамки елемента за допомогою CSS?
16. Як змінити колір тексту при наведенні за допомогою CSS?
17. Як змінити колір фону при наведенні за допомогою CSS?
18. Як змінити колір рамки при наведенні за допомогою CSS?
19. Як змінити розмір елемента при наведенні за допомогою CSS?

20. Як змінити позицію елемента при наведенні за допомогою CSS?
21. Як змінити вирівнювання тексту при наведенні за допомогою CSS?
22. Як змінити колір рамки при кліку за допомогою CSS?
23. Як змінити розмір елемента при кліку за допомогою CSS?
24. Як змінити позицію елемента при кліку за допомогою CSS?
25. Як змінити вирівнювання тексту при кліку за допомогою CSS?
26. Як змінити колір тексту при фокусуванні за допомогою CSS?
27. Як змінити колір фону при фокусуванні за допомогою CSS?
28. Як змінити колір рамки при фокусуванні за допомогою CSS?
29. Як змінити розмір елемента при фокусуванні за допомогою CSS?
30. Як змінити позицію елемента при фокусуванні за допомогою CSS?
31. Як змінити вирівнювання тексту при фокусуванні за допомогою CSS?
32. Як працює універсальний селектор \*?
33. Як працює селектор класу .?
34. Як працює селектор ідентифікатора #?
35. Як працюють селектори нащадків (div p)?
36. Як працює дочірній селектор (>)?
37. Як працюють псевдокласи (:hover, :focus)?
38. Як працюють псевдоелементи (::before, ::after)?
39. Що таке спадкування стилів?
40. Що таке opacity?

## Лабораторна робота №4 Основи JavaScript

### Мета

- Ознайомитись із основами мови JavaScript.
- Набути навички створення інтерактивних Web-сторінок.

### Завдання

1. Згідно варіанту додайте елементи інтерактивності до розробленого у попередніх лабораторних роботах Web-сайту.
2. Використовуйте змінні, функції, цикли та масиви для обробки даних.
3. Взаємодійте з DOM для динамічного оновлення контенту сторінки.

### Варіанти виконання завдань

1. Конвертер валют. Реалізуйте конвертацію валют за допомогою JavaScript.
2. Калькулятор. Створіть простий калькулятор з основними арифметичними операціями.
3. Гра «Вгадай число». Реалізуйте гру, де користувач повинен вгадати загадане число.
4. Таймер. Створіть таймер з можливістю встановлення часу.
5. Слайдер. Реалізуйте слайдер зображень з кнопками управління.
6. Форма зворотного зв'язку. Створіть форму з валідацією даних.
7. Генератор випадкових чисел. Реалізуйте генератор випадкових чисел у заданому діапазоні.
8. Список завдань. Створіть список завдань з можливістю додавання, видалення та редагування.

9. Калькулятор часу. Реалізуйте калькулятор для розрахунку часу між двома датами.

10. Генератор паролів. Створіть генератор паролів з можливістю вибору параметрів.

11. Гра «Камінь-ножиці-папір». Реалізуйте гру з вибором варіантів.

12. Генератор QR-кодів. Створіть генератор QR-кодів для введеного тексту.

13. Гра «Пам'ять». Реалізуйте гру на знаходження парних карток.

14. Генератор графіків. Створіть генератор графіків на основі введених даних.

15. Генератор таблиць. Реалізуйте генератор таблиць з можливістю редагування.

16. Генератор форм. Створіть генератор форм з можливістю вибору типів полів.

17. Генератор списків. Реалізуйте генератор списків з можливістю сортування.

18. Генератор текстів. Створіть генератор текстів на основі шаблонів.

19. Генератор зображень. Реалізуйте генератор зображень на основі введених параметрів.

20. Генератор анімацій. Створіть генератор анімацій з можливістю вибору параметрів.

21. Генератор звуків. Реалізуйте генератор звуків з можливістю вибору параметрів.

22. Генератор відео. Створіть генератор відео на основі введених параметрів.

23. Генератор аудіо. Реалізуйте генератор аудіо на основі введених параметрів.

24. Генератор текстур. Створіть генератор текстур на основі введених параметрів.

25. Генератор моделей. Реалізуйте генератор моделей на основі введених параметрів.



26. Генератор сцен. Створіть генератор сцен на основі введених параметрів.

27. Генератор об'єктів. Реалізуйте генератор об'єктів на основі введених параметрів.

28. Генератор персонажів. Створіть генератор персонажів на основі введених параметрів.

29. Генератор історій. Реалізуйте генератор історій на основі введених параметрів.

30. Генератор діалогів. Створіть генератор діалогів на основі введених параметрів.

31. Генератор квестів. Реалізуйте генератор квестів на основі введених параметрів.

32. Генератор рівнів. Створіть генератор рівнів на основі введених параметрів.

33. Генератор завдань. Реалізуйте генератор завдань на основі введених параметрів.

34. Генератор тестів. Створіть генератор тестів на основі введених параметрів.

35. Генератор опитувань. Реалізуйте генератор опитувань на основі введених параметрів.

36. Генератор анкет. Створіть генератор анкет на основі введених параметрів.

37. Генератор звітів. Реалізуйте генератор звітів на основі введених параметрів.

38. Генератор презентацій. Створіть генератор презентацій на основі введених параметрів.

39. Генератор документів. Реалізуйте генератор документів на основі введених параметрів.

40. Генератор сайтів. Створіть генератор сайтів на основі введених параметрів.

### **Вимоги до програмної реалізації**

- Змінні. Використовуйте змінні для зберігання даних.

- Функції. Створіть користувацькі функції для обробки даних.
- Цикли. Використовуйте цикли для повторення дій.
- Масиви. Використовуйте масиви для зберігання наборів даних.
- DOM. Взаємодійте з DOM для оновлення контенту Web-сторінки.

### **Критерії оцінювання**

1. Виконання вимог (40 балів):
  - використання змінних (10 балів);
  - застосування функцій (10 балів);
  - використання циклів (10 балів);
  - взаємодія з DOM (10 балів).
2. Якість коду (30 балів):
  - чистота та структурованість коду (10 балів);
  - відповідність стандартам W3C (10 балів);
  - наявність коментарів (10 балів).
3. Креативність (20 балів):
  - оригінальність дизайну (10 балів);
  - використання додаткових елементів (10 балів).
4. Документація (10 балів):
  - наявність коментарів у кодї (5 балів);
  - опис функціональності Web-сторінки (5 балів).

### **Додаткова інформація**

Мова JavaScript хоч і є C-подібною мовою програмування, але у ній наявні певні неочевидні, на перший погляд, особливості. Використання мовних моделей штучного інтелекту дозволить досить швидко та якісно виконати поставлені завдання. Отриманий код буде досить простим для розуміння у випадку попереднього вивчення мови C++.

## Контрольні запитання

1. Що таке JavaScript?
2. Які типи даних існують в JavaScript?
3. Як створити змінну в JavaScript?
4. Як створити функцію в JavaScript?
5. Як створити цикл в JavaScript?
6. Як створити масив в JavaScript?
7. Як змінити текст елемента за допомогою JavaScript?
8. Як змінити колір елемента за допомогою JavaScript?
9. Як змінити розмір елемента за допомогою JavaScript?
10. Як змінити позицію елемента засобами JavaScript?
11. Як змінити вирівнювання тексту засобами JavaScript?
12. Як змінити колір фону елемента засобами JavaScript?
13. Як змінити колір рамки елемента засобами JavaScript?
14. Як змінити колір тексту при наведенні засобами JavaScript?
15. Як змінити колір фону при наведенні засобами JavaScript?
16. Як змінити колір рамки при наведенні за допомогою JavaScript?
17. Як змінити розмір елемента при наведенні за допомогою JavaScript?
18. Як змінити позицію елемента при наведенні за допомогою JavaScript?
19. Як змінити вирівнювання тексту при наведенні за допомогою JavaScript?
20. Як змінити колір тексту при кліку засобами JavaScript?
21. Як змінити колір фону при кліку засобами JavaScript?
22. Як змінити колір рамки при кліку засобами JavaScript?

23. Як змінити розмір елемента при кліку засобами JavaScript?
24. Як змінити позицію елемента при кліку за допомогою JavaScript?
25. Як змінити вирівнювання тексту при кліку за допомогою JavaScript?
26. Як змінити колір тексту при фокусуванні за допомогою JavaScript?
27. Як змінити колір фону при фокусуванні за допомогою JavaScript?
28. Як змінити колір рамки при фокусуванні за допомогою JavaScript?
29. Як змінити розмір елемента при фокусуванні за допомогою JavaScript?
30. Як змінити позицію елемента при фокусуванні за допомогою JavaScript?
31. Як змінити вирівнювання тексту при фокусуванні за допомогою JavaScript?
32. Як змінити колір тексту при активному стані за допомогою JavaScript?
33. Як змінити колір фону при активному стані за допомогою JavaScript?
34. Як змінити колір рамки при активному стані за допомогою JavaScript?
35. Як змінити розмір елемента при активному стані за допомогою JavaScript?
36. Як змінити позицію елемента при активному стані за допомогою JavaScript?
37. Як працює for?
38. Як працює while?
39. Яка різниця між const, let і var?
40. Як додати новий елемент у масив?

## Рекомендована література

1. Двірничук К. В., Вацек Д. О. Веб-програмування та веб-дизайн : навч. посіб. Чернівці : ЧНУ ім. Ю. Федьковича, 2022. 472 с.

2. Босько В. В., Константинова Л. В., Марченко К. М., Улічев О. С. Web-програмування. Частина 1 (frontend) : навч. посіб. Кропивницький : ЦНТУ, 2022. 208 с.

3. Баран С. В. Основи web-програмування : навч. посіб. Кривий Ріг : ДУЕТ, 2023. 316 с.

4. Ціхоцький М. С., Лукічов В. В. Основи web-програмування. Теорія і практика : навч. посіб. Вінниця : ВНТУ, 2023. 128 с.

5. Решевська К. С. Комп'ютерні мережі та Web-програмування. Запоріжжя : ЗНУ, 2022. 101 с.

6. Хайрова Н. Ф., Петрасова С. В. Сучасні технології Web-програмування : навч. посіб. Харків : Панов А. М., 2020. 112 с.

7. DuRocher D. HTML and CSS QuickStart Guide. Albany : ClydeBank Media LLC, 2021. 364 p.

8. Troelsen A., Japikse P. Pro C# 10 with .NET 6. 11th ed. New York : Apress, 2022. 3053 p.

9. Noring C., Jain A., Fernandez M., Mutlu A., Jaokar A. AI-Assisted Programming for Web and Machine Learning, 1st ed. Birmingham : Packt Publishing, 960 p.

### *Корисні посилання*

1. Основи HTML & CSS українською для початківців – YouTube. URL: <https://www.youtube.com/playlist?list=PLenwk9TUJzJ6ep0oogRRQZH3PMtTGKqF1> (Last access: 13.12.2024).

2. Вивчення HTML українською / Уроки програмування HTML5 з нуля – YouTube. URL: <https://www.youtube.com/playlist?list=PL7vq4D0vOpQbLLkQTKJIV0rtkHtQ3Pses> (Last access: 13.12.2024).

3. Вивчення CSS українською / Уроки CSS3 для початківців з нуля – YouTube. URL: <https://www.youtube.com/playlist?list=PL7vq4D0vOpQZUc8vHgXKoF-GjXUIVbYT> (Last access: 13.12.2024).
4. Курс JavaScript українською / Уроки програмування JS з нуля – YouTube. URL: <https://www.youtube.com/playlist?list=PL7vq4D0vOpQbGerLS1pEgvPXSLUXgvFEo> (Last access: 13.12.2024).
5. The W3C Markup Validation Service. URL: [http://validator.w3.org/#validate\\_by\\_upload](http://validator.w3.org/#validate_by_upload) (Last access: 13.12.2024).
6. JavaScript Підручник. Основи програмування. Приклади. Уроки для початківців. W3Schools українською. URL: <https://w3schoolsua.github.io/js> (Last access: 13.12.2024).
7. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com>. (Last access: 13.12.2024).
8. HTML Reference. URL: <https://www.w3schools.com/tags/default.asp> (Last access: 13.12.2024).
9. Learn web development | MDN. URL: [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development](https://developer.mozilla.org/en-US/docs/Learn_web_development) (Last access: 13.12.2024).
10. CSS selectors - CSS: Cascading Style Sheets | MDN. URL: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors) (Last access: 13.12.2024).
11. ChatGPT. URL: <https://chatgpt.com/> (Last access: 13.12.2024).
12. Gemini. URL: <https://gemini.google.com/> (Last access: 13.12.2024).
13. Groq is Fast AI Inference. URL: <https://groq.com/> (Last access: 13.12.2024).
14. Chatbot Arena Leaderboard - a Hugging Face Space by lmarena-ai. URL: <https://huggingface.co/spaces/lmarena-ai/chatbot-arena-leaderboard> (Last access: 13.12.2024).