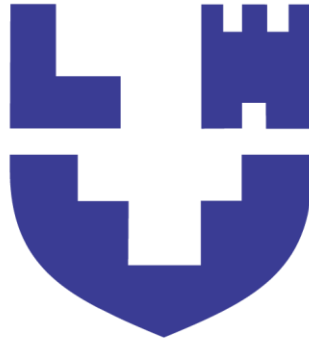


**Міністерство освіти та науки України
Луцький національний технічний університет**



Web-програмування

Методичні вказівки до лабораторних занять
для здобувачів першого (бакалаврського) рівня вищої освіти
освітньої програми «Цифровий маркетинг»
галузі знань 07 Управління та адміністрування
спеціальності 075 Маркетинг денної та заочної форм навчання

Луцьк 2024

УДК 339.138:004.4 (07)

W-37

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ
Директор бібліотеки _____ С. С. Бакуменко

Рекомендовано до видання вченою радою факультету бізнесу та права ЛНТУ,
протокол №__ від _____ 2024 року.

Голова вченої ради факультету бізнесу та права _____ Л. Л. Ковальська

Розглянуто і схвалено на засіданні кафедри маркетингу ЛНТУ, протокол №__ від
_____ 2024 року.

Завідувачка кафедри маркетингу _____ І. Ф. Лорві

Укладач : _____ О. М. Клімович, кандидат економічних наук, доцент
кафедри маркетингу ЛНТУ;

Рецензенти : _____ І. Ф. Лорві, кандидат економічних наук, доцент кафедри
маркетингу ЛНТУ;

Відповідальний

за випуск: _____ І. Ф. Лорві, кандидат економічних наук, доцент кафедри
маркетингу ЛНТУ.

W-37	Web-програмування : метод. вказівки до лабораторних занять для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань 07 Управління та адміністрування спец. 075 Маркетинг ОП Цифровий маркетинг денної та заоч. форм навч. / уклад. О. М. Клімович. Луцьк : Луцький НТУ, 2024. 24 с.
------	--

Методичне видання складене відповідно до діючої програми курсу «Web-програмування» та містить перелік тем лабораторних занять та перелік контрольних запитань, інструменти для написання коду та список рекомендованої літератури.

Призначене для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 075 Маркетинг освітньо-професійної програми «Цифровий маркетинг» денної та заочної форм навчання.

ЗМІСТ

<i>Змістовий модуль 1. Теоретичні основи Web-програмування</i>	4
Лабораторна робота 1. <i>Вступ до веб-програмування</i>	4
Лабораторна робота 2. <i>Основи HTML та CSS</i>	5
Лабораторна робота 3. <i>Основні компоненти веб-програмування</i>	10
Лабораторна робота 4. <i>Конструктори сайтів та CMS</i>	12
<i>Змістовий модуль 2. Веб-дизайн та SEO</i>	14
Лабораторна робота 5. <i>Основи веб-дизайну та UX/UI</i>	14
Лабораторна робота 6. <i>Основи SEO (Search Engine Optimization) просування</i>	15
Лабораторна робота 7. <i>Проектування та тестування веб-додатків</i>	18
Інструменти для написання коду	20
Список рекомендованих джерел	23

Лабораторна робота 1.

Як працювати з текстовим редактором:

1. Встановіть текстовий редактор: Завантажте та встановіть один з вищезгаданих текстових редакторів з офіційного сайту.
2. Створіть новий файл: Відкрийте редактор та створіть новий файл з розширенням .html, .css, або .js.
3. Напишіть код: Введіть код в редакторі та збережіть файл.
4. Відкрийте файл у браузері: Для HTML файлів просто відкрийте файл у браузері, щоб переглянути результат.

Життєвий цикл розробки веб-сайту

1. Планування: Визначення цілей, цільової аудиторії, аналіз конкурентів.
2. Дизайн: Створення макетів, прототипів (наприклад, у Figma).
3. Розробка: Написання коду для Frontend і Backend.
4. Тестування: Перевірка сайту на наявність помилок, забезпечення його безпеки та швидкодії.
5. Запуск: Розгортання сайту на сервері.
6. Підтримка та оновлення: Регулярне оновлення контенту, оптимізація роботи сайту.

Практична частина (демонстрація)

Створення найпростішої веб-сторінки:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Моя перша веб-сторінка</title>
</head>
<body>
  <h1>Вітаємо у світі веб-програмування!</h1>
  <p>Це ваш перший HTML-документ.</p>
</body>
</html>
```

Веб-програмування — це ключ до створення ефективної онлайн-присутності для бізнесу. Для маркетолога важливо не лише розуміти теоретичні основи, але й мати базові практичні навички роботи з веб-технологіями. Знання веб-програмування дозволяє покращити ефективність маркетингових кампаній, оптимізувати сайти для пошукових систем та аналітики.

Завдання для студентів:

Створити просту веб-сторінку, додати заголовок, короткий опис і посилання на будь-який сайт.

Контрольні питання

1. Що розуміють під поняттям веб-програмування та яку роль воно відіграє у цифровому маркетингу?
2. Які основні етапи розвитку веб-технологій можна виділити та як вони вплинули на сучасні онлайн-сервіси?
3. У чому полягає різниця між статичними та динамічними веб-сайтами?
4. Поясніть сутність клієнт-серверної архітектури веб-додатків.
5. Які функції виконує клієнтська частина веб-застосунку? Наведіть приклади.
6. Які завдання вирішує серверна частина веб-сайту або веб-додатка?

Лабораторна робота 2.

Тема: Основи HTML та CSS

Мета: Ознайомити студентів із базовими поняттями HTML і CSS, навчити їх створювати прості веб-сторінки, стилізувати їх за допомогою CSS та розуміти принципи побудови веб-інтерфейсів.

План заняття:

1. Що таке HTML і CSS?
2. Основи HTML: структура документа, основні теги.
3. Основи CSS: типи стилів, основні властивості.
4. Практичне завдання: створення простої веб-сторінки; Створення особистої сторінки профілю.

Нагадування!

• Розбір коду:

`<!DOCTYPE html>` – оголошує, що це HTML5-документ.

`<html lang="uk">` – кореневий тег, що містить увесь контент сторінки.

`<head>` – містить метайнформацію, назву сторінки, посилання на CSS.

`<body>` – містить видимий контент сторінки.

Тег	Опис
<code><h1></code>	Заголовок першого рівня
<code><p></code>	Параграф
<code><a></code>	Посилання
<code></code>	Зображення
<code><div></code>	Блоковий елемент
<code></code>	Вбудований елемент

Основні теги HTML

Тег	Опис
<code><h1></code> – <code><h6></code>	Заголовки різних рівнів (від найбільшого до найменшого).
<code><p></code>	Абзац тексту.
<code></code>	Гіперпосилання.
<code></code>	Зображення.
<code></code> / <code></code>	Невпорядкований/упорядкований список.
<code><table></code>	Таблиця.

Основи CSS

CSS дозволяє стилізувати HTML-елементи для покращення візуального вигляду веб-сторінки.

Основні властивості CSS

Властивість	Опис	Приклад
<code>color</code>	Колір тексту	<code>color: red;</code>
<code>font-size</code>	Розмір шрифту	<code>font-size: 20px;</code>
<code>background-color</code>	Колір фону	<code>background-color: yellow;</code>
<code>text-align</code>	Вирівнювання тексту	<code>text-align: center;</code>
<code>margin</code>	Відступ зовні	<code>margin: 10px;</code>
<code>padding</code>	Відступ всередині	<code>padding: 10px;</code>

Завдання для студентів:

Завдання 1.

Створіть веб-сторінку з заголовком, абзацом і посиланням.

1. Відкрийте **Visual Studio Code** або **Notepad++**.
2. Створіть файл `index.html` та додайте такий код:

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Лабораторна 2</title>
</head>
<body>
  <h1>Моя перша веб-сторінка</h1>
  <p>Це приклад роботи з HTML.</p>
  <a href="https://example.com">Перейти на інший сайт</a>
</body>
</html>

```

3. Збережіть файл та відкрийте його у браузері.

Завдання 2.

Додайте стилі для сторінки, щоб зробити її більш привабливою.

1. Створіть новий файл styles.css та напишіть код:

```
body {  
  background-color: #f0f0f0;  
  font-family: Arial, sans-serif;  
}  
  
h1 {  
  color: darkblue;  
  text-align: center;  
}  
  
p {  
  font-size: 18px;  
  color: #333;  
  text-align: justify;  
}
```

2. Підключіть CSS у index.html:

```
<head>  
  <meta charset="UTF-8">  
  <title>Лабораторна 2</title>  
  <link rel="stylesheet" href="styles.css">  
</head>
```

3. Збережіть зміни та оновіть сторінку у браузері.

Завдання 3.

Створення особистої сторінки профілю

- Створити просту HTML-сторінку з інформацією про себе.
- Застосувати базові стилі CSS для покращення вигляду сторінки.

Структура HTML:

Створіть файл profile.html з такою структурою:

```

2  <html lang="uk">
3  <head>
4    <meta charset="UTF-8">
5    <title>Моя сторінка профілю</title>
6    <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9    <header>
10     <h1>Привіт! Я – Марія</h1>
11   </header>
12
13  <main>
14    <section>
15      <h2>Про мене</h2>
16      <p>Я студентка, яка вивчає веб-програмування. Люблю подорожі, книги та каву ☕.</p>
17    </section>
18
19    <section>
20      <h2>Мої інтереси</h2>
21      <ul>
22        <li>Веб-дизайн</li>
23        <li>Цифровий маркетинг</li>
24        <li>Подорожі</li>
25      </ul>
26    </section>
27
28    <section>
29      <h2>Зв'яжіться зі мною</h2>
30      <p>Електронна пошта: <a href="mailto:maria@example.com">maria@example.com</a></p>
31    </section>
32  </main>
33
<footer>
  <p>© 2024 Марія. Усі права захищено.</p>
</footer>
</body>
</html>

```

Стилізація за допомогою CSS:
Створіть файл style.css та додайте стилі:

```
1  /* Загальні стилі */
2  body {
3      font-family: Arial, sans-serif;
4      background-color: #f0f0f0;
5      color: #333;
6      margin: 0;
7      padding: 0;
8      line-height: 1.6;
9  }
10
11 /* Заголовок */
12 header {
13     background-color: #0078d4;
14     color: white;
15     padding: 20px 0;
16     text-align: center;
17 }
18
19 /* Основний контент */
20 main {
21     width: 80%;
22     margin: 20px auto;
23     padding: 20px;
24     background-color: white;
25     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
26     border-radius: 10px;
27 }
28
29 /* Секції */
30 section {
31     margin-bottom: 20px;
32 }
33
34 /* Списки */
35 ul {
36     padding: 0;
37     list-style-type: square;
38 }
39
40 li {
41     margin: 5px 0;
42 }
43
44 /* Футер */
45 footer {
46     text-align: center;
47     padding: 10px 0;
48     background-color: #0078d4;
49     color: white;
50     margin-top: 20px;
51 }
52
53 /* Посилання */
54 a {
55     color: #0078d4;
56     text-decoration: none;
57 }
58
59 a:hover {
60     text-decoration: underline;
61 }
62
```

Очікуваний результат:

Після відкриття `profile.html` у браузері ви побачите:

- Заголовок із привітанням.
- Опис із короткою інформацією про себе.
- Список інтересів.
- Контактну інформацію.
- Стилізований фон, тіні та кольори.

Контрольні питання

1. Що таке HTML? Яку роль він відіграє у веб-програмуванні?
2. Які основні теги використовуються для структурування контенту?
3. Що таке CSS і як його можна підключити до HTML?
4. Яка різниця між `margin` та `padding`?
5. Як зробити текст червоним за допомогою CSS?

Висновки

- HTML створює структуру веб-сторінки, а CSS відповідає за її оформлення.
- Є три способи підключення CSS: `inline`, `internal`, `external`.
- CSS дозволяє змінювати кольори, шрифти, відступи, вирівнювання тексту та багато іншого.
- Використання окремого CSS-файлу допомагає підтримувати код у чистому вигляді.

Лабораторна робота 3

Тема: Основні компоненти веб-програмування (JavaScript, DOM, події)

JavaScript (JS) — мова, що дозволяє створювати динамічні та інтерактивні вебсторінки. Вона виконується на стороні клієнта (у браузері).

DOM (Document Object Model) — об'єктна модель документа, яка дозволяє JavaScript змінювати вміст, структуру та стилі HTML-документів.

✓ Завдання 1. Обробка події натискання кнопки

```
<button onclick="sayHello()">натисни мене</button>
<p id="result"></p>

<script>
  function sayHello() {
    document.getElementById("result").innerText = "Вітаємо на нашому сайті!";
  }
</script>
```

Змініть текст повідомлення на рекламне або інформаційне (наприклад: «*Запишіться на безкоштовну консультацію*»). Спробуйте замінити `innerText` на `innerHTML` і вставити тег `` у текст.

✓ Завдання 2. Введення та обробка даних

```
<input type="text" id="userName" placeholder="Ваше ім'я">
<button onclick="greetUser()">Привітати</button>
<p id="greeting"></p>

<script>
  function greetUser() {
    let name = document.getElementById("userName").value;
    document.getElementById("greeting").innerText = "Привіт, " + name + "!";
  }
</script>
```

Додайте перевірку: якщо поле порожнє — виводити повідомлення *"Будь ласка, введіть ім'я"*.

✓ Завдання 3. Зміна стилю елементів через JS

```
html

<p id="text">цей текст змінить свій стиль</p>
<button onclick="changeStyle()">Змінити стиль</button>

<script>
  function changeStyle() {
    let element = document.getElementById("text");
    element.style.color = "red";
    element.style.fontSize = "20px";
  }
</script>
```

Додайте другий стиль або кнопку «Повернути назад».

✓ Завдання 4. Обчислення значень у формі

```

<input type="number" id="price" value="100">
<input type="number" id="quantity" value="1">
<button onclick="calculate()">Обчислити</button>
<p id="total"></p>

<script>
  function calculate() {
    let price = document.getElementById("price").value;
    let qty = document.getElementById("quantity").value;
    let total = price * qty;
    document.getElementById("total").innerText = "До сплати: " + total + " грн";
  }
</script>

```

Додайте знижку 10% при кількості понад 5.

Контрольні питання

1. Що таке DOM і як він використовується у JS?
2. Як JavaScript може взаємодіяти з HTML-елементами?
3. У чому різниця між innerText та innerHTML?
4. Що таке подія у JavaScript? Назвіть приклади.
5. Як можна змінити CSS-стиль елемента за допомогою JS?

Лабораторна робота №4

Тема: Конструктори сайтів та CMS

Теоретичний вступ:

- **Конструктори сайтів** — це онлайн-платформи, що дозволяють створити сайт без програмування шляхом вибору шаблону та заповнення візуального інтерфейсу (Weblium, Wix, Squarespace).
- **CMS (Content Management System)** — більш гнучкі системи, які дозволяють керувати контентом через адмін-панель (WordPress, Joomla, Drupal).
- Використання цих систем дозволяє маркетологам швидко створити сайт, лендінг або блог для просування послуг чи товарів.

✓ Завдання 1. Створення сайту в конструкторі Weblium або Wix (чи використайте будь-який інший конструктор чи платформу створення сайтів)

Зареєструйтесь або увійдіть у свій обліковий запис.

Створіть новий сайт, обравши шаблон «Бізнес», «Маркетинг» або «Освіта».

Додайте до шаблону:

- заголовок сторінки;
- опис компанії або послуги;
- контакти;
- кнопку «Зв'язатися»;
- зображення або відео.

Опублікуйте сайт та збережіть лінк.

Рекомендація: Зробіть скриншот опублікованої сторінки і вставте його у звіт.

✓ Завдання 2. Ознайомлення з CMS (WordPress)

Перейдіть на wordpress.com

Створіть новий сайт (можна використати безкоштовний тариф).

Оберіть тему оформлення (наприклад, Astra, Twenty Twenty-Two).

Створіть такі сторінки:

- Головна;
- Про нас;
- Блог;
- Контакти.

Додайте зображення до головної сторінки та перший допис у блог.

Порада: Зверніть увагу на SEO-налаштування та адаптивність (чи коректно виглядає на телефоні).

✓ Завдання 3. Порівняння конструктора і CMS

Створіть таблицю-порівняння:

Критерій	Weblium / Wix	WordPress
Простота запуску	✓ дуже проста	• середній рівень
Гнучкість дизайну	• шаблонна	✓ більша свобода
SEO-можливості	• базові	✓ розширені плагіни
Швидкість редагування	✓ висока	• помірна
Потреба в хостингу	✗ не потрібно	✓ потрібно (якщо .org)

Контрольні питання

1. У чому основна відмінність конструктора сайтів і CMS?
2. Які функції потрібні маркетологу на сайті?
3. Що таке шаблон сайту і як його змінити?
4. Які переваги адаптивного дизайну?
5. Як обрати оптимальний інструмент для швидкого запуску рекламної сторінки?

Лабораторна робота 5.

Тема: Основи веб-дизайну та UX/UI

Інструменти:

- Онлайн-сервіс [Figma](#) або [Penpot](#);
- Веббраузер (рекомендовано Google Chrome);
- Акаунт Google або інший email для входу;
- Готовий контент: назва курсу, опис, фото, контактні дані (можна вигаданий).

Теоретичний вступ:

- **UX (User Experience)** — це загальний досвід користувача від взаємодії з продуктом. Включає зручність, логіку, доступність.
- **UI (User Interface)** — це зовнішнє оформлення інтерфейсу: кнопки, шрифти, кольори, іконки.
- Гарний інтерфейс — це той, що виконує функцію без зайвих дій і створює позитивне враження.

✓ Завдання 1. Аналіз прикладів хорошого та поганого UX

Перейдіть на галерею прикладів: <https://goodui.org/examples> або Behance. Оберіть **один сайт із хорошим UX** та **один з поганим** (через пошук або власний досвід).

Проаналізуйте за критеріями:

- логічна структура;
- читабельність;
- доступність СТА (кнопки дії);
- адаптивність.

У звіт вставити скріншоти з коротким коментарем.

✓ Завдання 2. Створення макету лендінгу у Figma

- Зайдіть на [figma.com](#) або [penpot.app](#)
- Створіть новий проєкт.
- Розробіть **прототип лендінгу** для навчального курсу або продукту (наприклад: «Цифровий маркетинг»).

Обов'язкові елементи:

- Заголовок та підзаголовок;
- СТА-кнопка «Записатись»;
- Зображення або ілюстрація;
- Блок відгуків користувачів;
- Форма зворотного зв'язку (email або телефон).

Форма має бути розміщена згідно з логікою користувацького потоку (user flow).

✓ Завдання 3. Рецензія на прототип колеги

Обміняйтесь посиланнями на прототип з іншим студентом.

Перегляньте макет і дайте письмовий фідбек:

- чи легко знайти головну кнопку?
- чи логічне розміщення блоків?
- які емоції викликає дизайн?

У звіті вказати ім'я колеги та висновки в 4–5 реченнях.

Контрольні питання

1. У чому різниця між UX і UI?
2. Які елементи обов'язкові для лендінгу?
3. Що таке СТА і як воно повинно виглядати?
4. Які принципи хорошого UX?
5. Назвіть 3 типові помилки у веб-дизайні.

Лабораторна робота 6.

Тема: Основи SEO (Search Engine Optimization) просування

Інструменти:

- Браузер Google Chrome;
- Інструменти аналізу SEO:
 - <https://neilpatel.com/seo-analyzer/>
 - <https://web.dev/measure/>
 - <https://pagespeed.web.dev/>
- Готові URL-адреси сайтів для аналізу.

Теоретичний вступ:

SEO (Search Engine Optimization) — це набір заходів, спрямованих на покращення видимості сайту у результатах пошуку Google та інших систем.

Ключові елементи:

- мета-теги (title, description);
- ключові слова;
- структура заголовків (h1, h2, ...);
- мобільна адаптивність;

- швидкість завантаження сторінки;
- якість контенту.

✓ Завдання 1. Аналіз SEO-структури обраного сайту

1. Виберіть **український сайт малого бізнесу** або сервісу (наприклад: школа англійської, онлайн-курс, локальний магазин).
2. Скопіюйте його головну URL-адресу.
3. Вставте адресу на <https://neilpatel.com/seo-analyzer/>.
4. Зверніть увагу на:
 - наявність title і meta description;
 - структуру заголовків (h1, h2);
 - ALT-тексти до зображень;
 - наявність помилок SEO.
5. Зробіть **скріншот результату** й додайте у звіт.

✓ Завдання 2. Перевірка швидкості та мобільної адаптивності

1. Вставте URL сайту в <https://pagespeed.web.dev/>.
2. Проаналізуйте показники:
 - швидкість завантаження для мобільного;
 - основні помилки;
 - адаптивність;
 - читабельність тексту та розміщення елементів.
3. Додайте оцінку з Google PageSpeed Insights до звіту.

✓ Завдання 3. Порівняння двох сайтів однієї тематики

1. Оберіть **2 сайти з однієї ніші** (наприклад: 2 курси англійської мови).
2. Проведіть аналіз SEO обох за допомогою одного з інструментів (наприклад: web.dev/measure).
3. Створіть таблицю порівняння:

Критерій	Сайт 1	Сайт 2
Title / Description	є / є	є / відсутній
ALT до зображень	є частково	відсутні
H1 / H2 структура	логічна	хаотична
Мобільна адаптивність	повна	часткова
Швидкість завантаження	85 / 100	52 / 100

✓ Завдання 4. Оцінка зручності та впливу SEO на маркетинг

- Оберіть сайт популярного бренду (наприклад: Work.ua, Allo, Comfy).
- Проаналізуйте з точки зору маркетолога:
 - Чи видно сайт у топ-результатах Google за ключовими словами?
 - Який вигляд має його сніпет (фрагмент у результатах пошуку)?
 - Чи зрозуміло з опису, що пропонується?

Напишіть короткий висновок: як SEO впливає на залучення користувачів.

✓ Завдання 5. Порівняння двох сайтів за SEO-параметрами

- Оберіть 2 сайти однієї тематики (наприклад, освітні послуги: Prometheus і EdEra).
- Проаналізуйте кожен через <https://web.dev/measure>.
- Зверніть увагу на:

Показник	Сайт А	Сайт В
SEO-оцінка		
Швидкість		
Мобільна оптимізація		
Доступність		
Поради інструменту		

Контрольні питання

1. Які елементи найбільше впливають на SEO?
2. У чому різниця між SEO-оптимізацією контенту та технічною оптимізацією?
3. Як швидкість сайту впливає на позицію в Google?
4. Чому важливо мати адаптивний дизайн?
5. Що таке мета-теги і як їх правильно заповнювати?

Лабораторна робота 7.

Тема: Проектування та тестування веб-додатків

Інструменти:

- Будь-який сучасний браузер (Chrome, Firefox);
- Демо-сайти або публічні вебдодатки (наприклад: Rozetka, Work.ua, Prom.ua, Zakaz.ua);
- Онлайн-сервіси UX-аналізу (опціонально):
 - <https://www.hotjar.com/>
 - <https://usabilityhub.com/>
 - <https://pagespeed.web.dev/>

Теоретичний вступ:

Вебдодаток — це інтерактивний сайт, що виконує функції програми (фільтрація, форма замовлення, особистий кабінет).

Юзабіліті-тестування (usability testing) — процес перевірки зручності використання інтерфейсу користувачем.

Під час тестування аналізується:

- швидкість виконання цільових дій;
- помилки;
- зручність навігації;
- емоційне сприйняття.

✓ Завдання 1. Користувацький сценарій (user flow)

1. Оберіть вебдодаток (наприклад: онлайн-магазин).
2. Визначте **ціль користувача**: купити товар, заповнити форму, зареєструватися.
3. Складіть послідовність дій від відкриття сайту до виконання цілі:
 - Перехід на головну → Пошук → Перехід до картки товару → Додавання в кошик → Оформлення замовлення.
4. Створіть **текстову або графічну схему**.

✓ Завдання 2. Оцінка зручності сайту

1. Зайдіть на сайт з мобільного телефону та з ПК.
2. Проаналізуйте такі критерії:
 - чи легко знайти потрібний розділ?
 - чи зрозуміло, що далі робити?
 - чи працюють кнопки / меню?
 - чи швидко завантажується?

3. Створіть таблицю:

Критерій	Оцінка	Коментар
Доступність головного СТА	4/5	Видно, але низький контраст
Швидкість	5/5	Завантажується до 2 секунд
Зрозумілість структури	3/5	Меню не очевидне на смартфоні

✓ Завдання 3. Мікротестування взаємодії

1. Оберіть 1 функціональний елемент (наприклад: форма пошуку або кошик).
2. Введіть тестові дані (наприклад, «куртка», або свій email).
3. Відслідкуйте:
 - Чи швидко реагує система?
 - Чи є повідомлення про помилки/успіх?
 - Чи можна легко повернутися на попередній крок?

Зробіть 2–3 скриншоти з описом.

Питання для самоконтролю:

1. Що таке user flow і як він допомагає UX-дизайнеру?
2. Чому важливо тестувати сайт з мобільних пристроїв?
3. Які основні показники зручності сайту?
4. Яка роль маркетолога у процесі UX-аналізу?
5. Які інструменти допомагають тестувати вебінтерфейси?

ІНСТРУМЕНТИ ДЛЯ НАПИСАННЯ КОДУ

1. Текстові редактори

Текстові редактори – це прості інструменти, які дозволяють писати і редагувати код. Вони підходять для початківців і часто використовуються професійними розробниками.

Популярні текстові редактори:

1. **Notepad++** (Windows): Простий та легкий текстовий редактор з підсвіткою синтаксису.
2. **Sublime Text** (Windows, Mac, Linux): Потужний текстовий редактор з багатьма можливостями.
3. **Visual Studio Code** (Windows, Mac, Linux): Безкоштовний редактор від Microsoft з підтримкою багатьох мов програмування та розширень.
4. **Atom** (Windows, Mac, Linux): Текстовий редактор від GitHub з багатьма можливостями налаштування.

Як працювати з текстовим редактором:

1. Встановіть текстовий редактор: Завантажте та встановіть один з вищезгаданих текстових редакторів з офіційного сайту.
2. Створіть новий файл: Відкрийте редактор та створіть новий файл з розширенням .html, .css, або .js.
3. Напишіть код: Введіть код в редакторі та збережіть файл.
4. Відкрийте файл у браузері: Для HTML файлів просто відкрийте файл у браузері, щоб переглянути результат.

2. Інтегровані середовища розробки (IDE)

IDE (Інтегровані середовища розробки) – це потужні інструменти, які надають функціональні можливості для написання, тестування та налагодження коду.

Популярні IDE:

- **Visual Studio Code:** Підтримує розширення для багатьох мов програмування та має вбудовані інструменти для налагодження.
- **WebStorm:** Потужна IDE від JetBrains спеціально для розробки веб-додатків.
- **Brackets:** Відкритий редактор коду з вбудованими функціями для веб-розробки.

Як працювати з IDE:

1. Встановіть IDE: Завантажте та встановіть IDE з офіційного сайту.
2. Створіть новий проект: Відкрийте IDE та створіть новий проект для веб-розробки.
3. Напишіть код: Використовуйте редактор коду для написання HTML, CSS та JavaScript.

4. Запустіть проект: Багато IDE мають вбудовані сервери для запуску проекту, що дозволяє легко переглядати зміни у браузері.

Приклад використання HTML, CSS та JavaScript разом

```
1 <!DOCTYPE html>
2 <html lang="uk">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Приклад сторінки</title>
7 <style>
8   body {
9     background-color: #f0f0f0;
10    font-family: Arial, sans-serif;
11  }
12  h1 {
13    color: #333;
14  }
15  p {
16    color: #666;
17  }
18 </style>
19 <script>
20   function showMessage() {
21     alert('Привіт, світ!');
22   }
23 </script>
24 </head>
25 <body>
26   <h1>Привіт, світ!</h1>
27   <p>Це мій перший документ з HTML, CSS та JavaScript.</p>
28   <button onclick="showMessage()">Натисни мене</button>
29 </body>
30 </html>
```

Результат

Привіт, світ!

Це мій перший документ з HTML, CSS та JavaScript.

Натисни мене

Привіт, світ!

Це мій перший документ з HTML, CSS та JavaScript.

Натисни мене



Присутня анімація при наведенні курсору.

Інструменти для роботи з веб-технологіями

Візуальні редактори:

- **VS Code** — популярний редактор коду для розробників.
- **Notepad++** — простий редактор для базових змін у коді.

Конструктори сайтів:

- **Wix, Tilda, Squarespace** — дозволяють створити сайт без знань програмування.
- **WordPress** — найпопулярніша CMS, що дозволяє керувати контентом без глибоких технічних знань.

Інструменти для аналітики:

- **Google Analytics** — для відстеження поведінки користувачів.
- **Google Tag Manager** — для управління маркетинговими тегами на сайті.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Семенюк І. І. Основи веб-програмування. HTML, CSS, JavaScript. – К.: Ліра-К, 2021.
2. IT-посібник «Маркетинг і цифрові технології» / під ред. О. Шестакової – Харків: УІМ, 2023.
3. Браян Гоган. HTML & CSS: Design and Build Websites. – Wiley, 2020.
4. Jon Duckett. JavaScript and jQuery: Interactive Front-End Web Development. – Wiley, 2021.
5. Кирилюк О. П. Цифровий маркетинг у вебсередовищі. – НУ «Львівська політехніка», 2022.
6. Котлер Філіп, Гермаван Катарджая, Іван Сетьяван. Маркетинг 4.0. Від традиційного до цифрового. Київ : КМ-БУКС, 2019. 224 с.

Онлайн-курси та платформи

7. Prometheus (UA): Основи веб-розробки, Цифровий маркетинг, HTML для початківців.
8. Coursera (EN/UA): Web Design for Everybody, Meta Front-End Developer Professional.
9. EdEra (UA): Вступ до програмування та HTML/CSS.
10. Udemy (EN): The Complete JavaScript Course, WordPress for Beginners.
11. Codecademy (EN): Інтерактивне навчання HTML, CSS, JavaScript, SEO.
12. GoITeens / GoIT UA: Курси для початківців із акцентом на UI/UX.

Документація та стандарти

13. [MDN Web Docs \(Mozilla\)](#) — офіційна документація HTML, CSS, JS
14. Google Search Central — ресурс для SEO-фахівців і маркетологів
15. [W3C](#) — стандарти веб-технологій
16. [Schema.org](#) — мікророзмітка для пошукових систем
17. Google Lighthouse — вимірювання швидкості, доступності, SEO

Українські освітні ресурси

18. Освіта Дія — безкоштовні мінікурси з IT та цифрової грамотності
19. ITVDN Україна — відеокурси з основ веб-програмування
20. [YouTube канал "Фрілансер по життю"] — україномовні практичні уроки HTML, CSS, JavaScript

Практичні інструменти та редактори

21. [Figma.com](#) — інструмент для UX/UI прототипів
22. [Visual Studio Code](#) — безкоштовний редактор коду
23. [Netlify](#) — безкоштовне хостування вебсайтів
24. [Glitch.com](#) — інтерактивне середовище для експериментів із кодом
25. PageSpeed Insights — перевірка продуктивності сайту

Web-програмування : метод. вказівки до лабораторних занять для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань 07 Управління та адміністрування спец. 075 Маркетинг ОП Цифровий маркетинг денної та заоч. форм навч. / уклад. О. М. Клімович. Луцьк : Луцький НТУ, 2024. 24 с.

Комп'ютерний набір О. М. Клімович

Редактор О. М. Клімович

Підп. до друку 2025 р.

Формат 60x84/16. Папір офс. Гарнітура Таймс.

Ум. друк. арк. 1,74. Обл.-вид. арк. 5,1.

Тираж 50 прим. Зам__.

Відділ іміджу та промоції

Луцького національного технічного університету

43018 м. Луцьк, вул. Львівська, 75

Друк – Відділ іміджу та промоції ЛНТУ