

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Реалізація web-сайту на основі HTML з використанням
JavaScript»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Боровик В.О.

Студента групи ІН-63

Шкурко А.В.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІН-63 спеціальності “Інформатика” денної форми навчання Шкурка Андрія Володимировича.

Тема: «Реалізація web-сайту на основі HTML з використанням JavaScript»

Затверджена наказом по СумДУ

№ _____ від _____ 2020 р.

Зміст пояснювальної записки: 1) Інформаційний огляд; 2) Постановка задачі; 3) Вибір методу рішення; 4) Програмна реалізація

Дата видачі завдання “ _____ ” _____ 2020 р.

Керівник випускної роботи _____ Боровик В.О.

Завдання прийняв до виконання _____ Шкурко А.В.

РЕФЕРАТ

Записка: 86 стор., 69 рис., 1 таблиця, 1 додаток, 17 джерел.

Об’єкт дослідження — веб-додаток для магазину гітар “GuitarLand”.

Мета роботи — розробка веб-додатку для магазину гітар “GuitarLand”.

Методи дослідження — в процесі програмної реалізації проекту було застосовано технології для побудови веб-додатків (HTML, CSS, JavaScript).

Результати — проведений аналіз літератури, після ознайомлення з існуючими методами та їх перевагами та недоліками було розроблено алгоритм реалізації веб-додатку. Створено веб-додаток для магазину гітар “GuitarLand”. Реалізовано повний функціонал для продажу товарів, а також можливість онлайн-оплати.

ЗМІСТ

Вступ.....	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1 Дослідження предметної області.....	6
1.2 Огляд існуючих аналогів.....	6
1.3 Постановка задачі.....	10
2 Вибір методу реалізації.....	10
2.1 Методи розроблення та конструювання структури веб-сайту.....	10
2.2 Використання текстів, списків, посилань та зображень	15
2.3 Види таблиць та використання форм	31
2.4 Використання каскадних таблиць стилів (CSS)	38
2.4.1. Приклади стилів:.....	39
2.4.2. Кольори та їх значення.....	41
2.4.3. Текст у CSS. Вказівка шрифту (<i>font-family</i>).	42
2.4.4. Використання блоків.	44
2.4.5. Форми, таблиці та списки.....	47
2.5 Об'єктна модель JavaScript.....	51
3 Практична реалізація.....	53
ВИСНОВКИ.....	60
СПИСОК ЛІТЕРАТУРИ.....	62
ДОДАТОК.....	64

Вступ

У наш час людина не може обійтися без інтернет-ресурсів, веб-технології дуже швидко поширюються у нашому житті. Можна спостерігати за потужним розвитком всесвітньої павутини, кожен прагне дізнатися все більше нового, та почерпнути шматок нових знань, або просто провести час. Інтернет-ресурси покращують швидкість обміну і засвоєння інформації, це сприяє тому, що вони становляться більш привабливими для користувачів.

Зараз у кожного є своя соціальна мережа, або своя сторінка в Інтернеті. Напевно, є ті, хто хотів би мати свій власний веб-сайт для різних цілей. Для кого це була би візитівка з інформацією про людину, а для когось картка банку, або медична картка.

Торгівельна сфера дуже швидко розвивається і кожен із магазинів прагне чимось виділитися. Для привернення уваги вони починають використовувати саме ці інтернет-ресурси. Магазины створюють свої сторінки в соціальних мережах та в цілому на просторі Інтернету.

Розробка інтернет-сайту для торгівельної сфери потребує залучення спеціалістів інформаційно-технічного напрямку. Не у кожного магазину є зайві кошти на залучення до роботи гарного спеціаліста. Саме утримання веб-сайту в мережі Інтернет потребує деяких фінансових затрат.

Отже, метою даної роботи є розробка онлайн-сервісу на основі сучасних веб-технологій, який буде задовольняти всі потреби магазину при мінімальних фінансових вливаннях.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Дослідження предметної області

Веб-сайтом називають елемент всесвітньої павутини. У ньому містяться веб-документи, об'єднані в єдину структуру. Веб-сайти мають свої адреси в Інтернеті та певну класифікацію.

Веб-ресурс – це сторінка (або набір сторінок), розміщених в мережі Інтернет, які можуть включати як текстову, графічну інформацію, так і мультимедіа-компоненти (відео, музику і т.д.) [13].

Сторінки веб-сайту можуть бути статичними і динамічними. Наприклад, вони можуть генеруватися на сервері гіпертекстовим препроцесором з використанням інформації з таблиць бази даних, або включати в себе інформацію, яку видає скрипт [6].

Унікальність адреси веб-сайту залежить від доменного імені. Для кожного сайту надається своя доменна адреса. Вона використовується для пошуку сторінки в мережі Інтернет.

1.2 Огляд існуючих аналогів

Тема роботи пов'язана зі створенням інтернет-магазину. Ідея ця не нова, в наш час існує безліч ресурсів, що мають схожу мету та функціонал. А отже, щоб створити конкурентноспроможний продукт, потрібно проаналізувати наявні аналоги та провести їх детальне вивчення. У даній роботі до розгляду та аналізу схожих рішень були обрані такі інтернет-магазини:

- «Muzline» (<https://muzline.ua/>) (рис. 1.1)
- «Gitara.in.ua» (<http://www.gitara.in.ua>) (рис. 1.2)
- «Muztorg» (<https://muztorg.ua/gitari/>) (рис. 1.3).

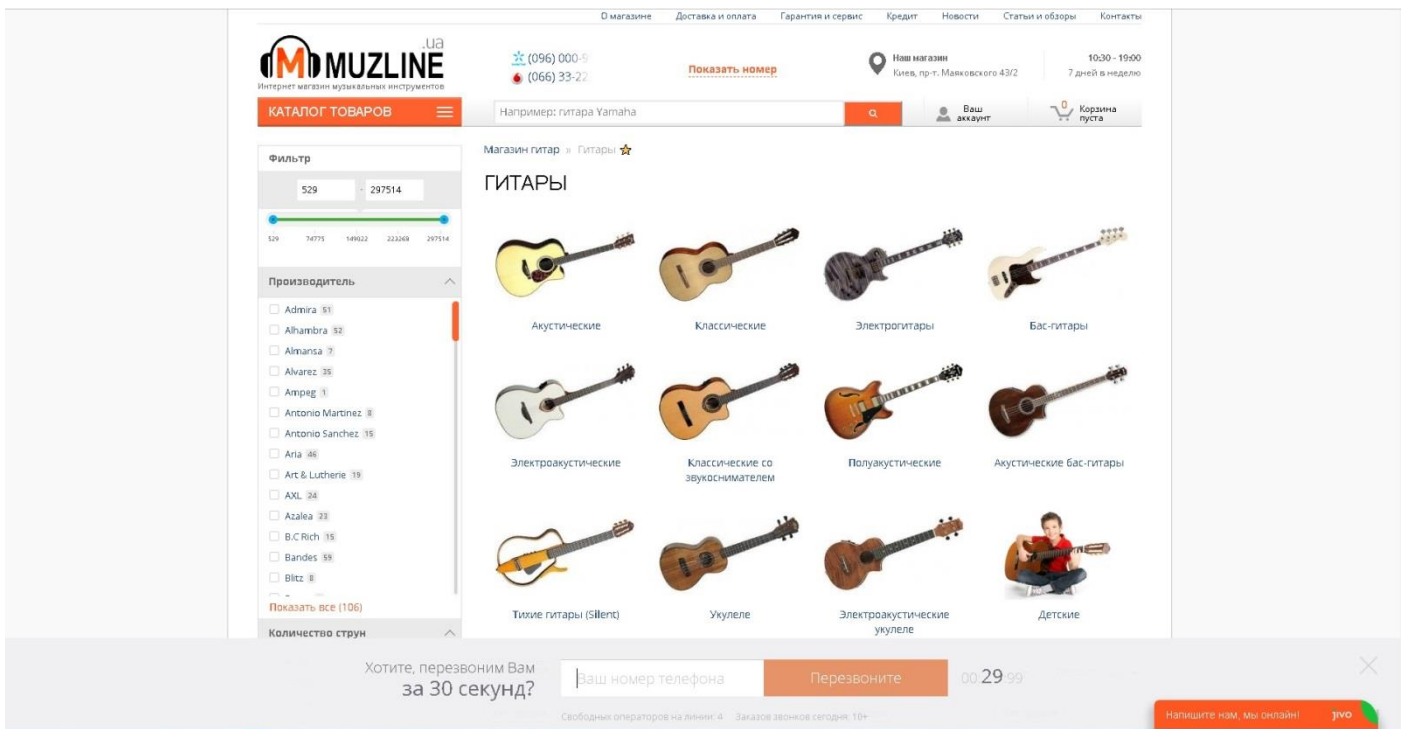


Рисунок 1.1 – Интернет-магазин Muzline

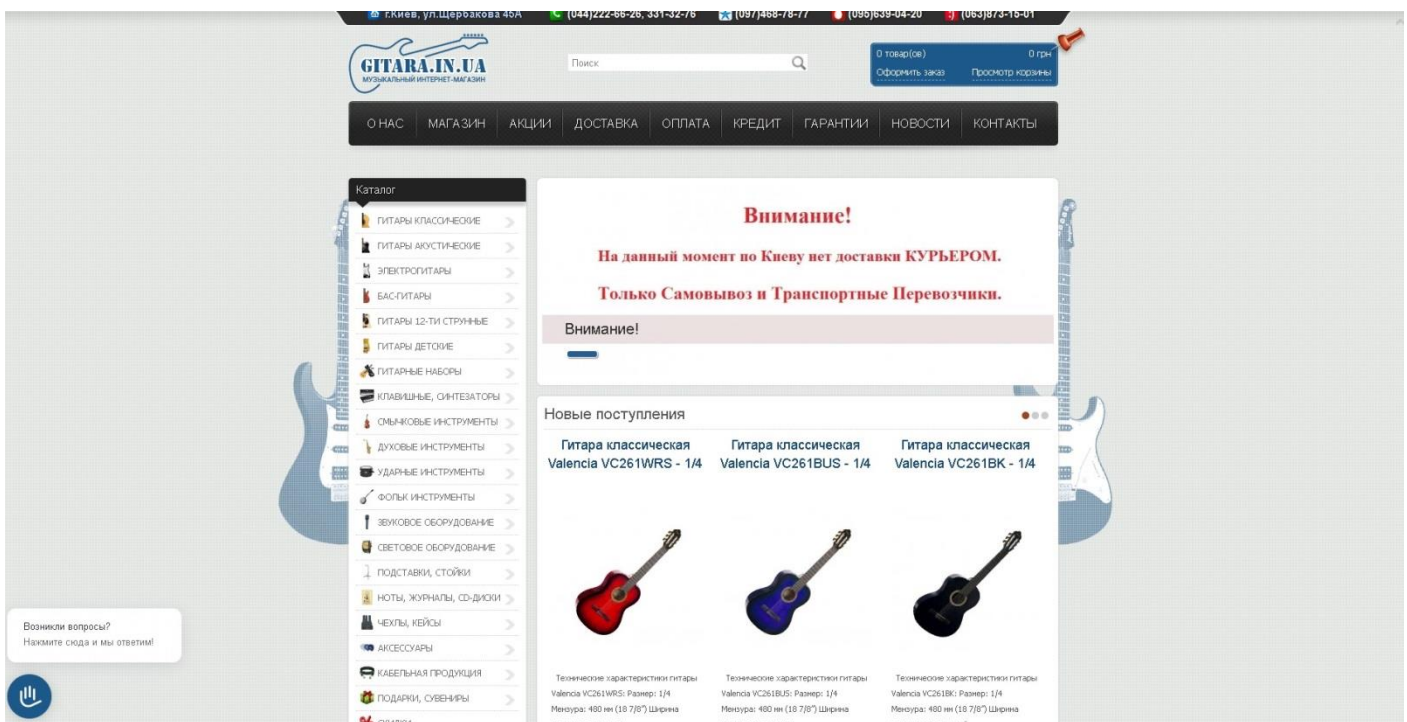


Рисунок 1.2 – Интернет-магазин Gitara.in.ua

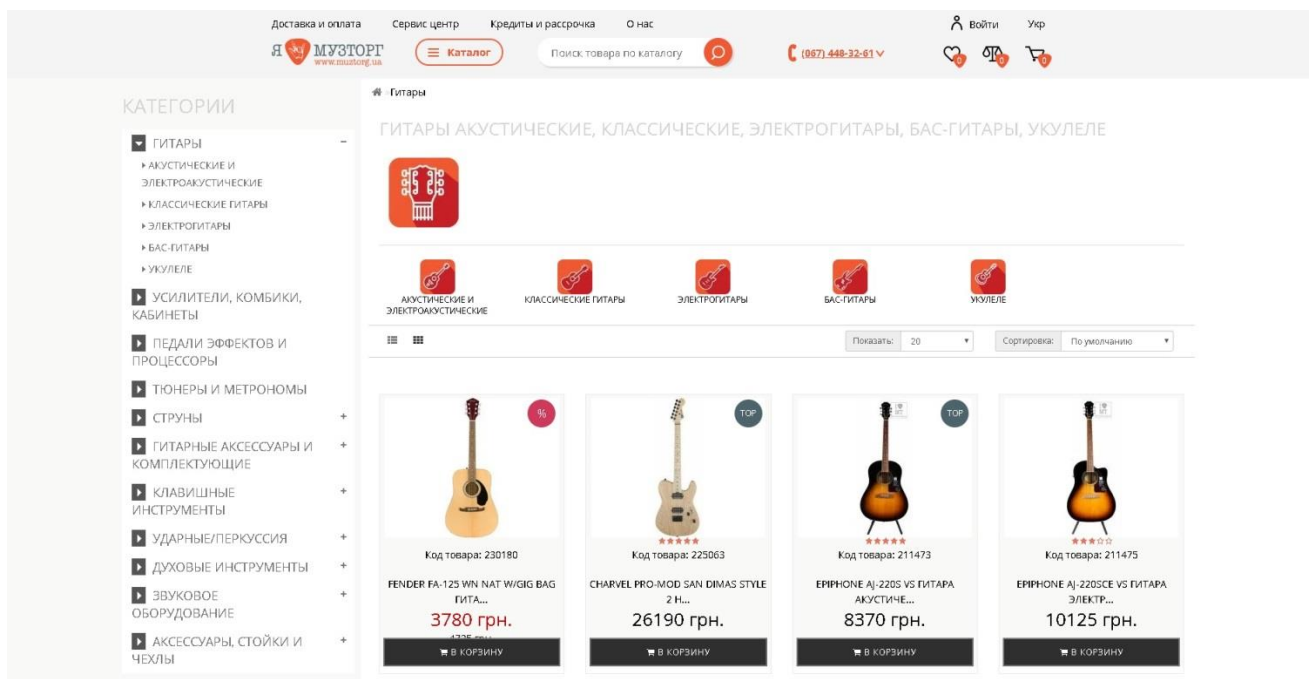


Рисунок 1.3 – Интернет-магазин Muztorg

В ході дослідження подібних веб-ресурсів та існуючих аналогічних рішень була побудована порівняльна таблиця (табл. 1.1).

Таблиця 1.1 – Порівняльна таблиця існуючих аналогів

Критерії	Назва сайту		
	“Muzline”	“Gitara.in.ua”	“Muztorg”
Зручність інтерфейсу	+	-	+
Сучасність дизайну	+	-	+
Адаптивність	+	+	+
Блок посиланнями на соціальні мережі	+	-	+
Відгуки клієнтів	+	+	+

Наявність особистого кабінету	+	-	+
Підписка для отримання спеціальних пропозицій	-	-	+
Контактна інформація	+	+	+

На основі аналізу сформованої порівняльної таблиці (табл. 1.1) виявлено переваги та недоліки існуючих web-сайтів.

Переваги:

- Зрозумілий для використання інтерфейс
- Адаптивна верстка
- Можливість залишити відгук
- Зворотній зв'язок.

Недоліки:

- Відсутність блоку з посиланнями на соціальні мережі
- Відсутність сортування продуктів
- Не зовсім вдалий дизайн.

Всі ці критерії мають бути враховані на наступному етапі нашої роботи, а саме – при формулюванні постановки задачі та при подальшій її реалізації.

1.3 Постановка задачі

Темою предсталеної роботи є верстка інтернет-магазину гітар. Програмний продукт повинен відповідати таким вимогам:

- Зрозумілий інтерфейс з сучасним дизайном
- Можливість реєстрації та авторизації
- Перегляд існуючих товарів (гітар)
- Використання декількох видів продуктів
- Перегляд додаткової інформації обраного та подібного товару
- Пошук гітари (по магазину)
- Можливість додавання продуктів до кошика покупця
- Можливість залишити відгук.

Результатом роботи повинен бути веб-додаток для продажу гітар, адаптивною версткою та сервером що буде забезпечувати виконання вище зазначених функцій та критерій.

2 Вибір методу реалізації

2.1 Методи розроблення та конструювання структури веб-сайту

Для конструювання сайту використовується програма Sublime Text 3. Багато веб-сторінок функціонують неначе електронні версії документів. Наприклад, газети публікують одні й ті ж репортажі на папері та на власних сайтах. За допомогою всесвітньої павутини можливо написати заяву на отримання паспорта. Інтернет-магазини містять онлайн-версії каталогів і засоби електронної комерції.

Структура важлива для всіх типів документів: вона допомагає читачам зрозуміти інформацію, яку ви намагаєтеся до них донести, а також сприяє більш простому переходу між частинами документів.

Статті, котрі публікуються в газетах повинні мати свій текст, заголовок, одне або більше зображень. Велика стаття може бути розділена на декілька частин підзаголовками, в ній також містяться спеціальні цитати.

Структура статті, що подається до всесвітнього перегляду буде лише незначно відрізнятись від тієї, що ви бачите на папері (такі публікації, наприклад, іноді мають будь-які аудіо- або відеофайли). Поглянувши на наступну сторінку, можна побачити відскановану журнальну або газетну статтю та її електронну версію, опубліковану на сайті журналу чи газети.

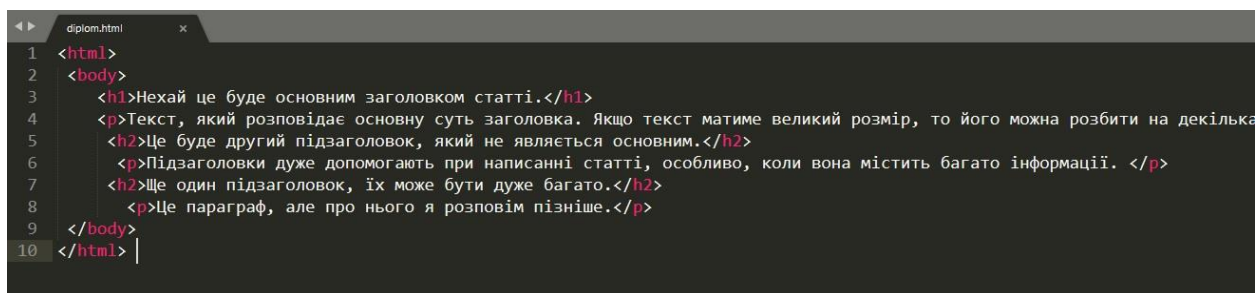
Можна також навести приклад, як виглядає інший вид документа. Візьмемо заяву на видачу паспорта. Як правило, подібна форма документа має кілька розділів, кожен з яких містить власний заголовок. Всі розділи мають поля для заповнення, які містять в собі питання до них, а також присутні комірочки, щоб ставити в них галочки або інші знаки. Структура цього документа, знову ж таки, дуже схожа на електронну версію форми, опубліковану на сайті [13, с. 18].

Ієрархію даних допомагає відобразити використання заголовків при складанні документів різних типів. Наведемо приклад будь-якого документа, який повинен починатися з заголовка, написаного великим шрифтом. Далі, як правило, подається вступ, яке містить найважливішу інформацію. В цю структуру можна додавати підзаголовки, що поміщаються в тексті нижче на сторінці. Найкраще було б при складанні документа за допомогою текстового редактора розділяти фрагменти тексту, для структурування їх. Підтеми можуть виділятися в окремі абзаци, крім того, у кожному підрозділі також може бути власний заголовок, що описує те, про що йдеться в тексті написаному нижче.

Отже, потрібно взяти простий документ Microsoft Word. Маємо змогу спостерігати, що у спадному списку показані різні стилі тексту, які можна застосувати в документі, наприклад, для створення заголовків різного рівня. Якщо людина часто працює з програмою Word, то вона, ймовірно, вже багато разів використовувала інструменти форматування або палітру кольорів для виділення заголовків і підзаголовків.

Для опису структури веб-документа необхідно додати трохи коду до тексту, який ви хочете відобразити на екрані.

Нижче на сторінці наведемо приклад коду на мові HTML. Зверніть увагу, що HTML-код надрукований червоним кольором, а текст, що виводиться на екран, - білим.



```
1 <html>
2 <body>
3   <h1>Нехай це буде основним заголовком статті.</h1>
4   <p>Текст, який розповідає основну суть заголовка. Якщо текст матиме великий розмір, то його можна розбити на декілька
5     <h2>Це буде другий підзаголовок, який не являється основним.</h2>
6     <p>Підзаголовки дуже допомагають при написанні статті, особливо, коли вона містить багато інформації. </p>
7     <h2>Ще один підзаголовок, їх може бути дуже багато.</h2>
8     <p>Це параграф, але про нього я розповім пізніше.</p>
9   </body>
10 </html>
```

Рис. 2.1 – Код із заголовками

Такий вигляд має структура статті на сервері. А тепер покажемо, як виглядає вона на веб-сторінці.

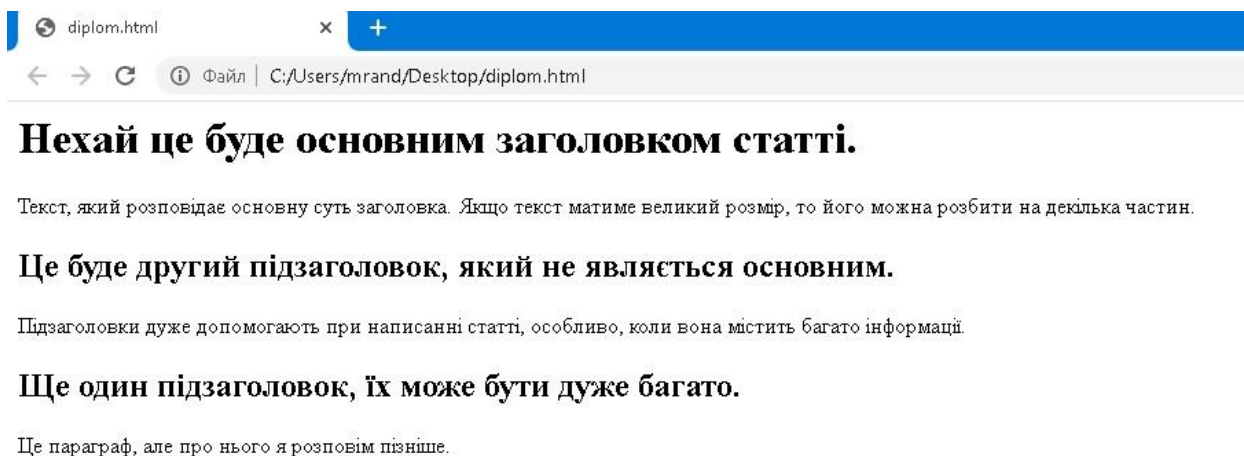


Рисунок 2.2 – Вигляд коду у браузері

Основний заголовок помітно відразу, бо він самого темного кольору, інші підзаголовки – світліші. З допомогою тегів можна вивести будь-який текст або статтю на екран вашого пристрою. Кожен тег відкривається і закривається. Ці теги розповідають нам про інформацію яка вказана поміж їх [13, с. 26].

Візьмемо тег `<html></html>` – він показує нам, що будь-яка інформація, що лежить між цими тегами являється HTML кодом.

Тег `<body></body>` – вказує на те, що все повинно бути розташоване в основному вікні браузера.

Текст поміж тегами `<h1></h1>` демонструє те, що це – основний заголовок.

Абзаци тексту повинні бути між тегами `<p></p>`.

Текст підзаголовку вказується тегами `<h2></h2>`.

Візьмемо `<p>`. Можна помітити, що елемент розташований між двома скобками. Тег, або, іншими словами елемент, вказує на абзац. У англійській мові «paragraph», а тому і взяли першу букву «P». Він являється елементом закритого типу, який закривається слешем «/», `</p>`.

Відкриваючі теги можуть мати в собі атрибути, які складаються з двох частин: ім'я та значення. Наприклад:

`<p lang = "ua" >` – тут текст українською;

`</p>` lang – ім'я атрибута;

ua – значення атрибута.

Ім'я атрибута показує, якого роду додаткова інформація, яку ви намагаєтесь донести.

Як приклад, у роботі використано атрибут lang, який слугує для позначення мови тексту цього елемента. Вміст елемента `<title>` – виводиться в заголовку вікна браузера (вище текстового поля, в яке, зазвичай, вводиться адреса сайту), або – в якості назви вкладки сторінки (якщо ваш Браузер використовує вкладки) [13].

HTML розшифровується як HyperText Markup Language – мова розмітки гіпертексту. В цьому випадку слово «гіпертекст» означає, що за допомогою мови HTML ви можете створювати посилання, що дозволяють відвідувачам швидко і легко

переміщатися з однієї сторінки на іншу. Також мова розмітки дозволяє створювати примітки до тексту, а ці примітки, в свою чергу, надають додаткове значення вмісту документа. Уявімо, що є веб-сторінка: до вихідного тексту, який ви хочете відобразити у вікні браузера додається якийсь код, який використовується браузером для коректного відображення сторінки. Тому теги, що додаються вами, називаються «елементами розмітки» [13].

Розглянемо детально створення нової, своєї веб-сторінки. Для цього потрібно виконати ряд наступних дій. На комп'ютері під керуванням операційної системи Windows, потрібно запуснути програму Блокнот (або іншими словами Notepad). Це можна зробити наступним чином: на екрані Пуск (Start) в операційній системі Windows 7, 8, 10 клацнути правою кнопкою миші і обрати пункт «Всі програми» при запуску (All Programs) в правому нижньому кутку екрана. У списку додатків в групі стандартні – Windows (Accessories-Windows) клацнути мишею по ярлику програми Блокнот (Notepad). У попередніх версіях операційної системи вікна, вибрати команду меню Пуск - Всі програми - Стандартні - Блокнот. Також можна завантажити безкоштовний текстовий редактор Notepad++ з офіційного сайту. Якщо ви користуєтеся більш ранньою версією Windows, виконайте команду меню Пуск – введіть у вікні програми код [13, с. 35].

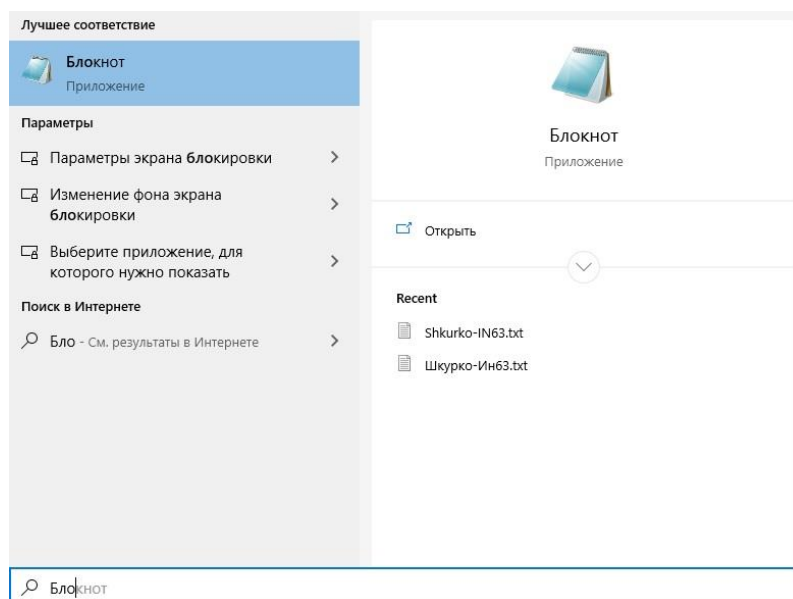


Рисунок 2.3 – Вигляд блокноту

У меню «файл (File)» виберіть команду «зберегти як (Save As)». Збережіть файл і запам'ятайте, де він знаходиться. Збережіть цей файл під будь-яким ім'ям, але в форматі html. Не забудьте вибрати пункт «всі файли (All Files)» у спадному списку тип «файлу (Save as type)», щоб зберегти документ як веб-сторінку. Для веб-сторінок з кириличними літерами обов'язково виберіть пункт UTF-8 у спадному списку кодування (Encoding). Це необхідно для правильного відображення російського, або українського тексту у вікні браузера. Запустіть браузер. У меню файл (File) виберіть команду «відкрити (Open)». Перейдіть в папку, що містить тільки що створений файл, виділіть його і клацніть по кнопці «відкрити (Open)» [13, с. 36].

2.2 Використання текстів, списків, посилань та зображень

При верстці веб-сторінки *текст* розмічають тегами. Теги надають тексту додаткове значення і дозволяють браузерам правильно відображати структуру сторінки.

У цьому розділі потрібно загострити увагу на тому, як додавати розмітку до тексту. А тому, виділимо такі підтеми:

- Структурна розмітка
- Семантична розмітка.

Вважаємо, що більш правильно розпочати із заголовків. Всього існує 6 видів заголовків (h1, h2, h3, h4, h5, h6). Вони поділяються від більшого масштабу до меншого. Наведемо приклад кожного з цих заголовків, використавши блокнот.

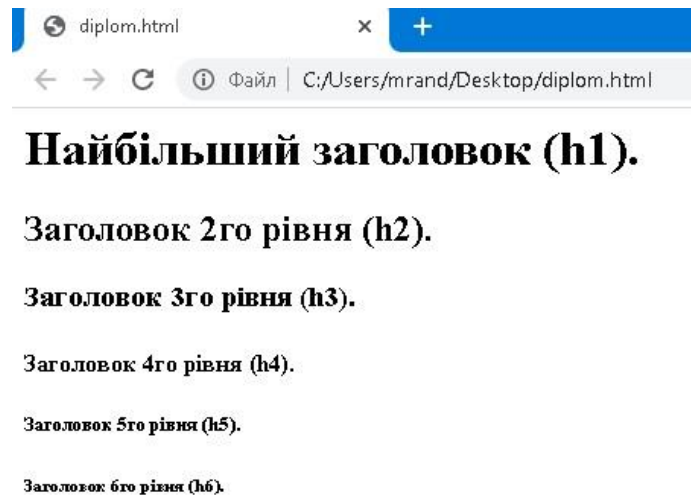


Рисунок 2.4 – Демонстрація заголовків в браузері

Розібравшись із заголовками, перейдемо до абзаців. Вони являються невід’ємною частиною будь-якої статті. Отже, що потрібно знати про абзаци. Їх можна легко побачити в HTML-кодї, вони позначаються літерою «P» (<p> </p>). Для того, щоб створити абзац, потрібно ввести текст між тегами. Якщо стоїть декілька тегів поспіль, браузер за замовчуванням відокремить їх один від одного. Нижче наведемо приклад [1].

```
1 <html>
2 <body>
3   <p>Візьму звичайний текст, нехай це буде першим абзацем. </p>
4   <p>Створю інший абзац, який буде відокремлено від першого за замовчуванням браузера.</p>
5 </body>
6 </html>
```

Рисунок 2.5 – Код абзаців

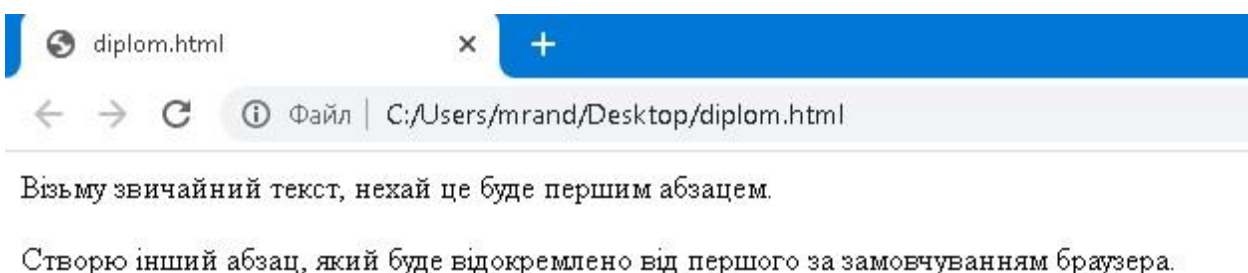
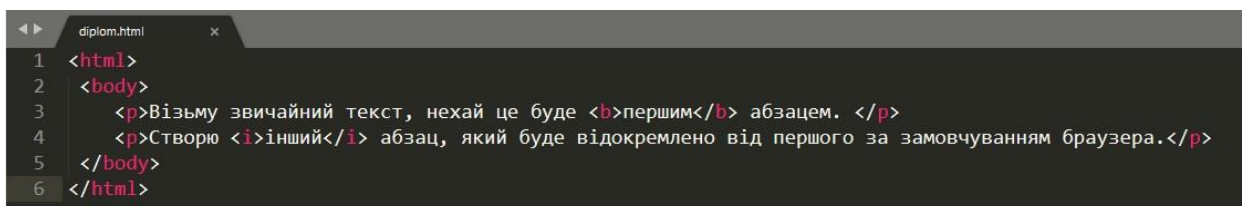


Рисунок 2.6 – Демонстрація коду абзаців

Можемо спостерігати, що текст відокремлений.

Щоб зробити текст, або слово жирним шрифтом потрібно використати тег ``, для написання слова курсивом – тег `<i></i>`. Вони існують для того, щоб підкреслити (виділити) в тексті важливе слово, або декілька слів. Візьмемо той же код, але зробимо декілька виправлень [5].



```
1 <html>
2 <body>
3 <p>Візьму звичайний текст, нехай це буде <b>першим</b> абзацем. </p>
4 <p>Створю <i>інший</i> абзац, який буде відокремлено від першого за замовчуванням браузера.</p>
5 </body>
6 </html>
```

Рисунок 2.7 – Абзац з виділеними словами

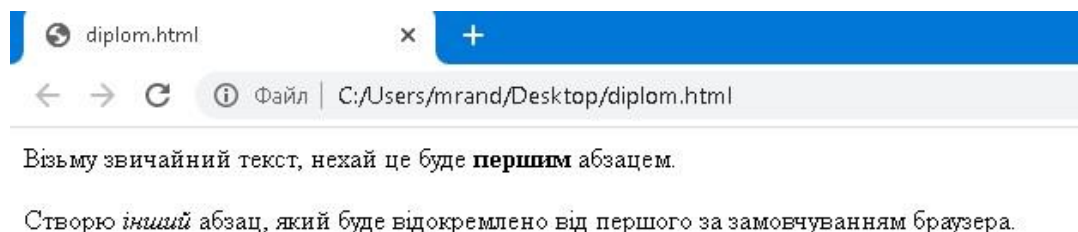


Рисунок 2.8 – Демонстрація коду в браузері слів

У даному прикладі використано тег `` на слово «першим» і воно на сторінці відобразилось жирним шрифтом. Слово «інший» стоїть між відкриваючим (`<i>`) і закриваючим (`</i>`) тегом, а тому його виділено курсивом.

Звернемо увагу на такі теги як `` і ``. Вони допомагають в математиці, фізиці та інших точних науках. Тег *sup* використовують для піднесення до степеня (або, іншими словами, його називають надрядковий тег), а тег *sub* вважають підрядковим тегом, бо символ зноситься під літеру збоку [5]. Наведемо приклади:

```
diplom.html x
1 <html>
2 <body>
3 <p>E = mc<sup>2</sup>. </p>
4 <p>H<sub>2</sub>O.</p>
5 </body>
6 </html> |
```

Рисунок 2.9 – Формула із хімії та фізики



Рисунок 2.10 – Формула у браузері

У першому та другому випадку використано різні теги на цифру 2. В першому випадку, використавши тег *sup* її піднято, в другому з тегом *sub* – опущено.

Важливо також знати про таку властивість браузера, як згортання пробілів. Верстальники веб-сторінок часто додають додаткові пробіли в код, або навіть поміщають деякі елементи на нові рядки з метою підвищення читабельності коду [6].

Коли браузер зустрічається з двома або більше пробілами, що йдуть один за одним, відображається тільки один. Аналогічно відбувається і з перенесенням рядка, сприймається одиничний символ пробілу [6].

Досить часто можна зіткнутися з тим, що творці веб-сторінок користуються згортанням пробілів, щоб розставити необхідні відступи у вихідному коді сторінок для підвищення читабельності [6].

Приклад:

```
1 <html>
2 <body>
3 <p>Немає різниці скільки відступів буде між цими словами . </p>
4 <p>Я можу написати текст ось так.</p>
5
6 <p> Або ось так пишу цей текст </p>
7
8
9
10
11 </body>
12 </html>
```

Рисунок 2.11 – Код згорання пробілів

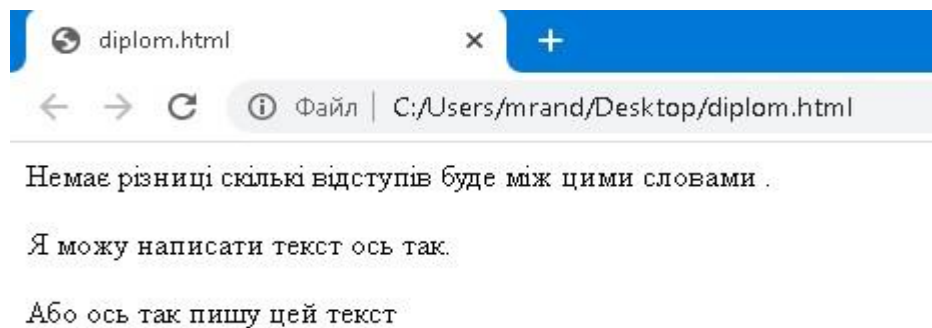


Рисунок 2.12 – Вигляд згорання пробілів у браузері

Тег `
`. Як було показано вище, браузер автоматично починає кожен абзац з нового рядка. Однак можна додати переклад рядка в межах будь-якого абзацу, для цього знадобиться тег `
`.

Тег `<hr />`. Щоб візуально розділити теми книги, за допомогою тега `<hr />` можна додати в текст горизонтальну лінію. У мові HTML існує кілька елементів, що не містять тексту між відкриваючими і закриваючими тегами. Їх прийнято називати порожніми елементами. Написання таких тегів відрізняється від стандартного. Зазвичай порожній елемент складається тільки з ОДНОГО тега, перед правою, що закриває дужкою якого вставляються пробіл і слеш. Деякі верстальники веб-сторінок опускають пробіл і слеш, хоча їх вказівка і рекомендується для проходження специфікації [6, с. 51].

Приклад:

```
diplom.html x
1 <html>
2 <body>
3 <p>Візьмемо текст, який ми хочемо<br /> перенести на інший рядок. </p>
4 <p>Візьмемо текст, який ми хочемо<hr /> підкреслити.</p>
5 </body>
6 </html> |
```

Рисунок 2.13 – Теги в кодї

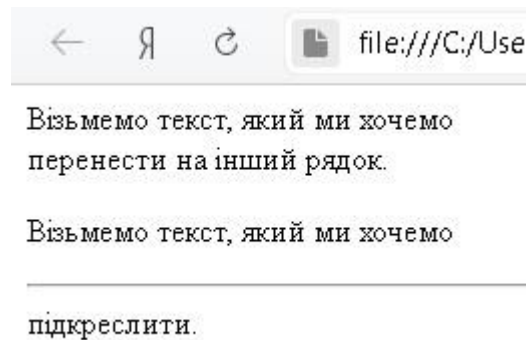


Рисунок 2.14 – Демонстрація тегів у браузері

Важлива інформація: Елемент `` використовується для позначення того, що текст, поміщений в нього, має високий ступінь важливості. Наприклад текст цього елемента може бути інтонаційно виділений при читанні програмою екранного доступу. За замовчуванням браузери відображають вміст елемента `` шрифтом з напівжирним накресленням [13].

Логічний наголос ``. Елемент `` використовується для позначення логічного наголосу, яке дещо змінює значення всього речення. За замовчуванням браузери відображають вміст елемента `` шрифтом з курсивним накресленням [1].

```
diplom.html x
<html>
<body>
<p><strong>Елемент strong</strong> використовується для позначення того, що текст, поміщений в нього, має високий ступінь важливості. </p>
<p><em>Елемент em</em> використовується для позначення логічного наголосу, яке дещо змінює значення всього речення.</p>
</body>
</html>
```

Рисунок 2.15 – Елементи `` та ``

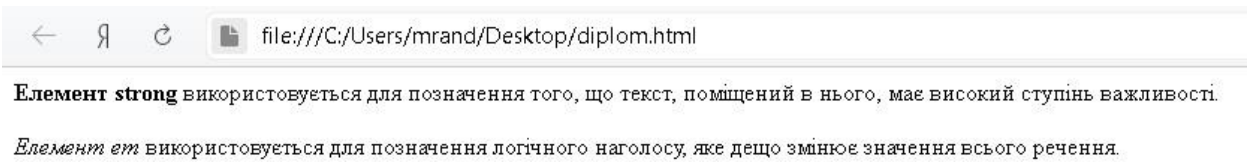


Рисунок 2.16 – Вигляд елементів на сторінці

Цитати. Для створення цитат в мові HTML використовуються два елементи:

1) Елемент `<blockquote>` використовується для відображення довгих цитат, що займають цілий абзац. Треба звернути увагу на те, що всередині елемента `<blockquote>` необхідно використовувати і елемент `<p>`. Здебільшого всі браузери схильні до виділення вмісту елемента `<blockquote>` відступами, однак цей елемент не варто використовувати тільки для того, щоб створити фрагмент тексту з відступом, краще реалізувати таке за допомогою каскадних таблиць стилів (CSS) [6, с. 74].

2) Елемент `<q>` прийнято використовувати для більш коротких цитат, що поміщаються всередині абзацу тексту. Спочатку передбачалося, що браузері будуть укладати вміст елемента `<q>` в лапки, тому багато хто уникає використовувати даний елемент. Для обох елементів може бути вказаний атрибут *cite*, який вказує, з якого сайту запозичена та чи інша цитата. Значення цього атрибута – URL-адреса сайту, що містить більш детальну інформацію про джерело цитування [13].

Приклади:

```
1 <html>
2 <body>
3   <blockquote>
4     <p> Елемент blockquote використовується для відображення довгих цитат, що займають цілий абзац. </p>
5   </blockquote>
6   <p> Елемент q прийнято використовувати для більш <q>коротких цитат</q>, що поміщаються всередині абзацу тексту. </p>
7 </body>
8 </html>
```

Рисунок 2.17 – Теги `<blockquote>` та `<q>`

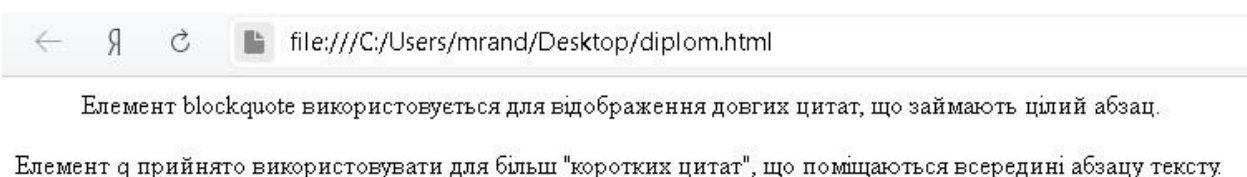


Рис. 2.18 – Перевірка тегів на сторінці

Отже, розібравшись з цитатами, можна перейти до *аббревіатур*. Для цього знадобиться тег `<abbr>`. При вживанні в тексті будь-якої аббревіатури слід скористатися тегом `<abbr>`. У відкриваючому тезі можна використовувати атрибут *title* для вказівки розшифровки скорочення [1].

У мові HTML4 для опису акронімів реалізований спеціальний елемент `<acronym>`. Для відображення розшифровки акроніма використовується атрибут *title* (аналогічний атрибуту елемента `<abbr>`). У мові HTML5 і для аббревіатур, і для акронімів використовується тег `<abbr>` [5].

```
1 <html>
2 <body>
3   <p><abbr title = "Кілометри">Км</abbr> </p>
4   <p><acronym title = "Кілометри за годину"> Км/год </acronym> </p>
5 </body>
6 </html>
```

Рисунок 2.19 – Аббревіатури в кодї

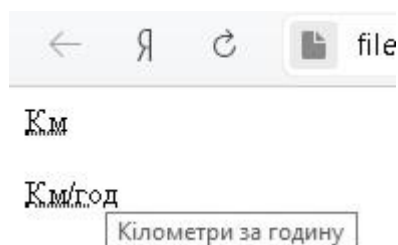


Рисунок 2.20 – Аббревіатури в браузері

Також одним із важливих тегів є виправлення контенту. Це закреслення та підкреслення слів чи речень. За допомогою тегів `<ins>` (підкреслює) та `` (закреслює). Елемент `<s>` дозволяє позначити помилкове слово чи декілька слів [5].

```
1 <html>
2 <body>
3   <p> Я хочу <del>закреслити</del> це слово, а це слово <ins>підкреслити</ins>. </p>
4   <p> І нехай це буде <s>помилкове слово</s>, яке нам не потрібне.</p>
5 </body>
6 </html>
```

Рисунок 2.21 – виправлення контенту

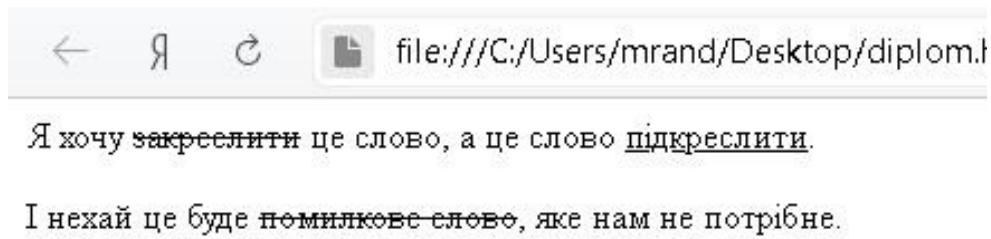


Рисунок 2.22 – Відображення виправлення контенту в браузері

Списки в HTML поділяють на 5 типів:

- 1) Упорядковані списки (нумеровані).
- 2) Неупорядковані списки (маркеровані).
- 3) Список визначень.
- 4) Вкладений список.
- 5) Багаторівневий нумерований список.

Упорядковані списки створюються за допомогою елемента ``. Кожен елемент списку поміщається між відкриваючим тегом `` і закриваючим ``. За замовчуванням браузері автоматично розставляють відступи для елементів списків.

```
diplom.html x
1 <html>
2 <body>
3 <ol>
4 <li>Microsoft</li>
5 <li>Google</li>
6 <li>Apple</li>
7 <li>IBM</li>
8 </ol>
9 </body>
10 </html> |
```

Рисунок 2.23 – Таблиці HTML



Рисунок 2.24 – Демонстрація таблиць

Неупорядковані (маркеровані). Для того, щоб зробити неупорядкований, список потрібно використати елемент ``. Для цього типу списків підходить така сама система, як і до упорядкованих списків. Потрібно вказати теги `` між елементам `` ``.

```
diplom.html x
1 <html>
2 <body>
3 <ul>
4 <li>Microsoft</li>
5 <li>Google</li>
6 <li>Apple</li>
7 <li>IBM</li>
8 </ul>
9 </body>
10 </html> |
```

Рисунок 2.25 – Таблиці з маркером

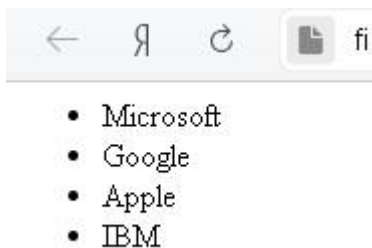


Рисунок 2.26. – Демонстрація таблиць з маркером

Список визначень <dl>. Список визначень створюється за допомогою елемента <dl> і, як правило, містить набір понять і їх визначень. Усередині елемента <dl> використовують пари елементів <dt> і <dd>.

<dt> використовується, щоб записати поняття чи термін.

<dd> використовується для визначення поняття.

Іноді можна зіткнутися зі списками, в яких два поняття мають одне визначення, або, де для одного поняття наводиться кілька визначень.

```
1 <html>
2 <body>
3 <dl>
4 <dt>Поети:</dt>
5 <dd>Василь Олександрович Сухомлинський</dd>
6 <dd>Володимир Миколайович Сосюра</dd>
7 <dt>Прозаїки:</dt>
8 <dd>Леся Українка</dd>
9 <dd>Богдан Михайлович Стельмах</dd>
10 <dd>Володимир Кирилович Винниченко</dd>
11 </dl>
12 </body>
13 </html>
```

Рисунок 2.27 – Список визначень (код)

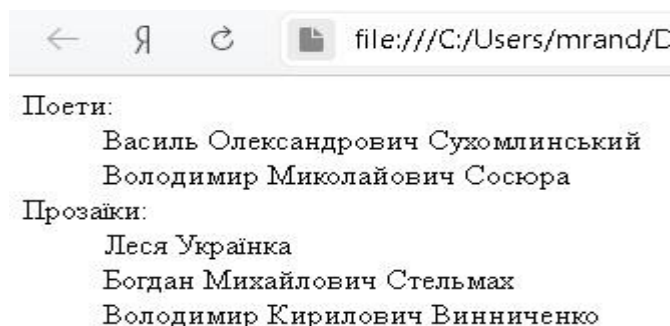


Рисунок 2.28 – Список визначень (браузер)

Вкладений список. Досить часто можливостей простих списків не вистачає. Наприклад, при створенні змісту ніяк не обійтися без вкладених пунктів. Усередині елемента допустимо створення вкладеного списку, або списку другого рівня [10].

Браузери відображають вкладені списки з великим відступом в порівнянні з основним, або батьківським, списком. Крім того, браузер, як правило, також змінює вигляд маркерів вкладеного неупорядкованого списку [10, с. 66].

```
diplom.html
1 <html>
2 <body>
3 <ul>
4 <li>Пункт 1.</li>
5 <li>Пункт 2.
6   <ul>
7     <li>Підпункт 2.1.</li>
8     <li>Підпункт 2.2.
9       <ul>
10        <li>Підпункт 2.2.1.</li>
11        <li>Підпункт 2.2.2.</li>
12      </ul>
13    </li>
14    <li>Підпункт 2.3.</li>
15  </ul>
16 </li>
17 <li>Пункт 3.</li>
18 </ul>
19 </body>
20 </html>
```

Рисунок 2.29 – Вкладений список (код)

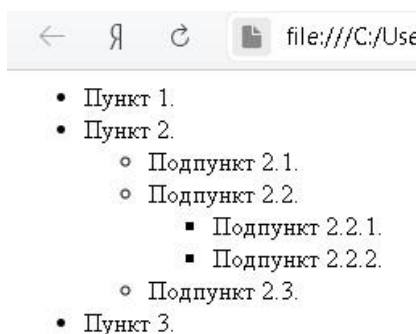


Рисунок 2.30 – Вкладений список (браузер)

Багаторівневий нумерований список використовується для відображення елементів списку на різних рівнях з різними відступами. Така розмітка за замовчуванням створює для кожного вкладеного списку нову нумерацію, що починається з одиниці [10].

HTML-посилання створюються за допомогою елементів <a>, <area> і <link>. Посилання являють собою зв'язок між двома ресурсами, одним з яких є поточний документ [1]. Посилання можна поділити на дві категорії:

1) Посилання *на зовнішні ресурси* створюються за допомогою тега `<link>` і використовуються для розширення можливостей поточного документа при обробці браузером;

2) *Гіперпосилання* – посилання на інші ресурси, які користувач може відвідати або завантажити.

Звернемо увагу на структуру посилань. Гіперпосилання створюються за допомогою парного тега `<a>< / a>`. Всередину тега поміщається текст, який буде відображатися на веб-сторінці. Текст посилання відображається в браузері з підкресленням, колір шрифту – Синій, при наведенні на посилання курсор миші змінює вигляд [1].

Обов'язковим параметром тега `<a>` є атрибут *href*, який задає URL-адресу веб-сторінки.



```
diplom.html
1 <html>
2 <body>
3 <a href="http://google.com">Google</a>
4 </body>
5 </html>
```

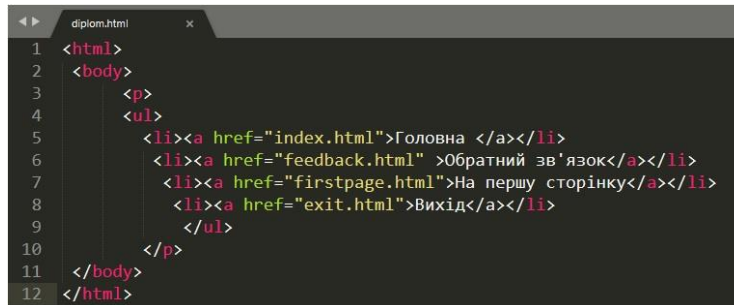
Рисунок 2.31 – Посилання в HTML



Рисунок 2.32 – Вигляд посилань в браузері

Посилання на інші сторінки сайту. При створенні посилань на інші сторінки сайту немає необхідності вказувати в URL-адресі його доменне ім'я. Замість цього можна скористатися скороченим варіантом, іменованим також відносною URL-адресою. Якщо всі сторінки сайту знаходяться в одній і тій же папці, то значенням атрибута *href* має бути лише ім'я файлу сторінки. Якщо сторінки сайту розподілені по різних

папках, то доведеться скористатися дещо складнішим синтаксисом для вказівки місця розташування потрібної сторінки щодо поточної. Якщо поглянути на код для завантаження прикладів сторінок для кожної з глав, то побачимо, що на сторінці `index.html` поміщені посилання з вказівкою відносних URL-адрес [13].



```
1 <html>
2 <body>
3   <p>
4     <ul>
5       <li><a href="index.html">Головна </a></li>
6       <li><a href="feedback.html" >Обратний зв'язок</a></li>
7       <li><a href="firstpage.html">На першу сторінку</a></li>
8       <li><a href="exit.html">Вихід</a></li>
9     </ul>
10  </p>
11 </body>
12 </html>
```

Рисунок 2.33 – Посилання на інші сторінки

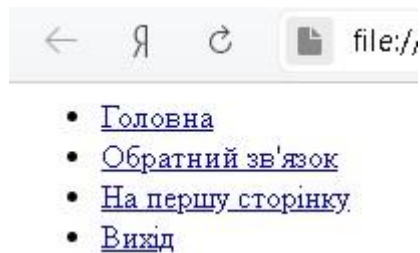


Рисунок 2.34 – Вигляд посилань на інші сторінки у браузері

Відносні URL-адреси можна використовувати при створенні посилань на інші сторінки сайту. Вони являють собою короткий варіант вказівки браузеру розташування необхідних файлів [13].

Із папки в папку. Для створення посилання на файл, розташований в тій же самій папці, вкажіть ім'я цього файлу (більше нічого вказувати не потрібно).

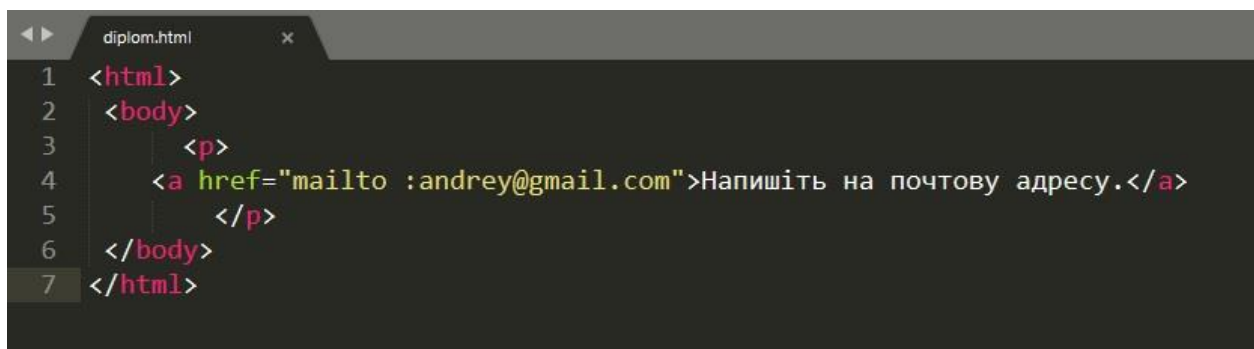
Дочірня папка. Для створення посилання на файл з дочірньої папки вкажіть ім'я папки, слеш, а потім ім'я потрібного файлу [13].

Внучата папка. Вкажіть імена дочірньої папки, внучатої папки і потрібного файлу, розділені слешами [13].

Батьківська папка. Введіть `.. /` для вказівки, що файл знаходиться в папці одним рівнем вище, після чого вкажіть ім'я потрібного файлу [13].

Прабатьківська папка. Повторіть введення символів .. / щоб показати, що ви хочете піднятися на два рівні вгору (а не на один), після чого вкажіть ім'я потрібного файлу.

Посилання на електронну пошту *mailto*. Для створення посилання, яке запускало б програму електронної пошти, встановлену на комп'ютері відвідувача, і дозволяло б відправити електронний лист на вказану адресу, потрібно скористатися елементом `<a>`. Однак тепер значення атрибута *href* має починатися зі слова *mailto:*, після якого необхідно вказати адресу електронної пошти, на яку відвідувач повинен відправити лист. На малюнку праворуч можна побачити посилання на електронну пошту, яка виглядає як звичайне посилання, але при клацанні мишею по ньому запускається програму для відправки електронної пошти і створюється новий лист, в адресному рядку якого буде введено адресу, вказану при створенні посилання [13].



```
1 <html>
2 <body>
3   <p>
4     <a href="mailto :andrey@gmail.com">Напишіть на поштову адресу.</a>
5   </p>
6 </body>
7 </html>
```

Рисунок 2.35 – Посилання на електронну пошту

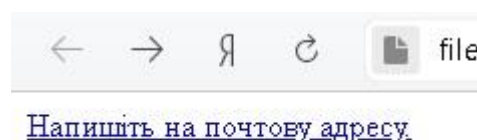


Рисунок 2.36 – Посилання на електронну пошту (браузер)

Існує досить багато підстав, щоб додати на веб-сторінку будь-яке *зображення*. Це може бути, наприклад, логотип, фотографія, ілюстрація, схема або діаграма. Перш ніж додавати зображення на сайт, потрібно задуматися про декілька моментів, але

час, витрачений на них, окупиться тим, що сайт буде виглядати набагато більш привабливо [5].

Якщо розробляти сайти з нуля, потрібно використовувати спеціальні папки для зручнішого використання зображення. Якщо сайт буде рости і розширюватися, розміщення всіх зображень в окремій папці спростить управління файлами. При створенні великого сайту можна додати в папку *images* кілька підпапок. Наприклад, такі зображення, як логотипи і кнопки, можна зберігати в підпапці *interface*, фотографії товарів – в підпапці *products*, а зображення, пов'язані з новинами сайту – в підпапці *news* [5].

Як саме додавати зображення до сайту? Для цього потрібно використати тег `` разом з атрибутом *src* (вказує на місце розташування зображення). Також існують атрибути *alt* (добавляє опис зображення), *title* (добавляє титул) [5].

Наприклад:

```

```

Найчастіше використовується елемент `` з ще двома атрибутами, що визначають його розмір:

- *Height* – даний атрибут встановлює висоту зображення в пікселях.

- *Width* – даний атрибут встановлює ширину зображення в пікселях.

Як правило, час завантаження зображень значно довше, ніж HTML-коду, що встановлює всю іншу розмітку сторінки. Тому рекомендується вказувати розміри зображень, щоб браузер міг продовжити обробку тексту сайту, не чекаючи закінчення їх завантаження і залишивши необхідну кількість порожнього місця. Все частіше творці сайтів встановлюють розміри зображень за допомогою каскадних таблиць стилів {CSS}, а не за допомогою мови HTML [5].

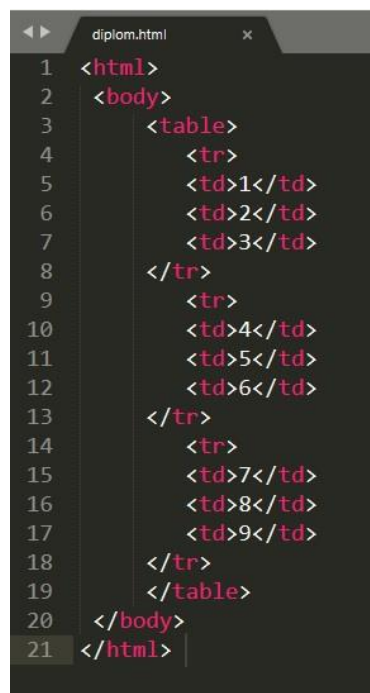
2.3 Види таблиць та використання форм

Існує деяка кількість видів інформації, яку слід відображати у вигляді сітки або *таблиці*. Наприклад результати спортивних змагань, інформація фондових ринків, розклади поїздів. Для створення таблиць потрібно мати уявлення про спеціальні терміни, такі як рядки і стовпці. Що таке таблиця і як її використовувати [13]?

Таблиця представляє інформацію у вигляді сітки. Прикладами таблиць можуть служити також фінансові звіти, телепрограми і результати спортивних змагань. Сітка допомагає сприймати великі обсяги даних шляхом співвіднесення їх з підписами по осях [1].

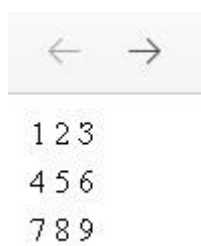
Окремий блок сітки називається *коміркою* таблиці. Для того, щоб створити таблицю треба використати основний тег `<table>`. Тег `<tr>` позначає початок нового рядка таблиці. Після нього поміщаються елементи `<td>`, кожен з яких відповідає окремій комірці в цьому рядку. Кінець рядка позначається закриваючим тегом `</tr>`.

Кожна комірка таблиці повинна бути представлена елементом `<td>*`. Кінець комірки позначається закриваючим тегом `</td>`. Деякі браузери автоматично створюють лінії навколо таблиць та/або комірок. Приклад:



```
1 <html>
2   <body>
3     <table>
4       <tr>
5         <td>1</td>
6         <td>2</td>
7         <td>3</td>
8       </tr>
9       <tr>
10        <td>4</td>
11        <td>5</td>
12        <td>6</td>
13      </tr>
14      <tr>
15        <td>7</td>
16        <td>8</td>
17        <td>9</td>
18      </tr>
19    </table>
20  </body>
21 </html>
```

Рисунок 2.37 – Приклад таблиці



←	→	
1	2	3
4	5	6
7	8	9

Рисунок 2.38 – Демонстрація вигляду таблиці

Елемент `<th>` використовується так само, як і елемент `<td>`, однак його призначення – створення заголовка рядка або стовпця. Слід пам'ятати, що, навіть якщо комірка не містить даних, то все одно необхідно представити її порожнім елементом `<td>` або `<th>`, в іншому випадку таблиця буде некоректно відображена в браузері (у прикладі перша комірка першого рядка порожня). Вживання елемента `<th>` допомагає людям, що використовують програми екранного доступу, покращує результативність індексування таблиць пошуковими системами, а ще цей елемент допомагає управляти оформленням таблиць при використанні CSS. Для позначення того, чи відноситься даний заголовок до рядка або стовпця, можна скористатися атрибутом *scope* елемента `<th>`. Цей атрибут здатний приймати одне з двох значень: *row* або *col* для позначення, до чого саме (до рядка або стовпця) відноситься даний заголовок. Як правило, браузери виділяють напівжирним шрифтом вміст елемента `<th>` і вирівнюють його по центру комірки [1].

У мові HTML існують три елементи, які допомагають відрізнити основний контент сторінки від першого і останнього рядків (які іноді містять принципово різну інформацію). Ці елементи корисні людям, які використовують програми екранного доступу, а крім того, вони допомагають виділяти дані розділи таблиці особливим стилем (що можна навчитися робити, почавши вивчати CSS) [3].

Заголовки таблиці повинні бути поміщені в елемент `<thead>`.

Основний вміст (тіло) таблиці має знаходитися всередині елемента `<tbody>`.

Останній, завершальний рядок таблиці повинен бути вказаний всередині елемента `<tfoot>`.

За замовчуванням дуже мало браузерів виділяють вміст цих трьох елементів, проте веб-дизайнери змінюють їх зовнішній вигляд за допомогою каскадних таблиць стилів (CSS) [3]

Розглянемо використання форм. Розробниками мови HTML концепція форми була запозичена для позначення набору елементів, що дозволяють збирати дані, що вводяться відвідувачами сайтів. HTML-форми надають весь необхідний набір елементів для збору даних користувача, незалежно від того, чи додається просте поле введення пошукового запиту, чи створюється складний додаток для заповнення заяви на отримання паспорта [13].

Структура форми `<form>`. Всі елементи форми поміщаються в тег `<form>`. Для нього завжди повинен бути вказаний атрибут *action*, крім того, для цього елемента часто встановлюють атрибути *method* і *id* [13]. Кожен елемент `<form>` вимагає вказівки атрибута *action*, значенням якого є URL-адреса сторінки-отримувача введених даних при відправці форми на сервер [13].

Method форми можна відправити одним з двох методів: *get* або *post*. При використанні методу *get* введені дані будуть додані в кінець URL-адреси, зазначеної в атрибуті *action*.

Введення тексту `<input>`. Елемент `<input>` використовується для створення компонентів форми декількох типів. Значення атрибута *type* вказує цей тип.

Type = "text". Коли атрибуту *type* присвоюється значення *text*, це означає, що буде створено однорядкове поле введення тексту.

Name. При введенні користувачем даних у веб-форму сервер повинен знати, які відомості відносяться до якого елемента форми (наприклад, при обробці форми авторизації на сайті сервер повинен знати, що було введено в якості імені користувача, а що - в якості пароля). Тому для кожного елемента форми слід

вказувати атрибут *name*. Значення цього атрибута служить для ідентифікації елемента форми і відправляється на сервер разом з введеними даними [13].

Maxlength. Можна використовувати атрибут *maxlength* для обмеження кількості символів, які відвідувачеві дозволяється ввести в дане поле. Наприклад, при запиті року ви можете обмежити кількість символів, встановивши значення атрибута *maxlength* рівним 4 [13].

Size. При створенні нових форм не слід вказувати атрибут *size*. Він використовувався раніше для встановлення ширини поля введення тексту (в якості одиниці виміру виступала кількість видимих символів). Так, наприклад, значення 3 створювало поле введення тексту, ширина якого була достатньою для відображення трьох введених символів (хоча користувачі при бажанні могли ввести і більшу кількість). Для управління шириною елементів нових форм рекомендується використовувати засоби каскадних таблиць стилів (CSS). Атрибут *size* згадано тільки тому, що зустрітися з ним можна при перегляді вихідного коду старих сайтів [13].

Приклад:

Браузер

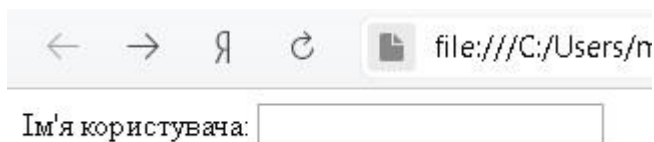


Рисунок 2.39 – Форма в HTML

Введення пароля проводиться аналогічно до введення тексту, але з використанням іншого типу *type= "password"*. Коли атрибуту *type* присвоюється значення *password*, на сторінці створюється однорядкове поле введення тексту, яке функціонує як звичайне поле введення тексту, з тією лише різницею, що символи які вводяться в нього маскуються. Це зроблено, щоб не допустити витоку таких важливих даних, як пароль, у разі, якщо хтось підглядає з-за спини відвідувача сайту.

Атрибут *name* встановлює ім'я поля введення пароля, що відправляється на сервер разом з паролем, введеним користувачем [13].

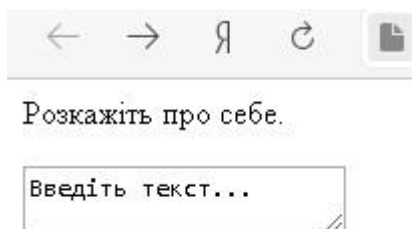
Приклад:



A screenshot of a web browser window showing a registration form. The browser's address bar contains the file path `file:///C:/Users/mi`. Below the address bar, there are two input fields. The first field is labeled "Ім'я користувача:" and the second field is labeled "Пароль:". Both fields are empty.

Рисунок 2.40 – Стандарте меню реєстрації

Текстова область. Елемент `<textarea>` використовується для створення багаторядкових областей введення тексту. На відміну від інших елементів форми, він не є порожнім, а отже, для нього слід вказати теги відкриття і закриття. Текст, поміщений між ними, буде виведений в поле при завантаженні сторінки. Якщо відвідувач не видалить цей текст, то він відправиться на сервер разом з введеними ним даними. На деяких сайтах реалізований сценарій на мові JavaScript, що очищає текстову область при клацанні по ній мишею [1].



A screenshot of a web browser window showing a text area form element. The browser's address bar contains the file path `file:///C:/Users/mi`. Below the address bar, there is a text area with the placeholder text "Розкажіть про себе." and a text input field with the placeholder text "Введіть текст...".

Рисунок 2.41 – Текстова область

Перемикачі `<input>`.

Тип `type = "radio"` перемикачі дозволяють відвідувачам сайту вибрати тільки один варіант із запропонованих.

Атрибут *name* відправляється серверу разом зі значенням обраного перемикача. Коли відповіддю на питання є вибір одного з декількох варіантів, всі перемикачі, що формують список відповідей, повинні мати однакове значення атрибута *name* [5].

Атрибут *value* встановлює значення обраного перемикача, що відправляється серверу. Значення кожного перемикача має бути унікальним всередині групи (щоб сервер знав, який варіант відповіді вибрав відвідувач) [5].

Атрибут *checked* вказує, який із запропонованих варіантів повинен бути обраний за замовчуванням при завантаженні сторінки (якщо це необхідно). Атрибут приймає єдине значення *checked*. Даний атрибут повинен бути встановлений тільки у одного перемикача з групи.

Приклад:

```
1 <html>
2 <body>
3 <form action="http://www.priklad.com/profile.php">
4 <p>Виберіть вид спорту, в який грають ногами: <br />
5 <input type="radio" name="genre" value="foot" checked="checked" /> Футбол
6 <input type="radio" name="genre" value="bask" /> Баскетбол
7 <input type="radio" name="genre" value="voll" /> Волейбол
8 </p>
9 </form>
10 </body>
11 </html>
```

Рисунок 2.45 – Перемикачі одиничного вибору

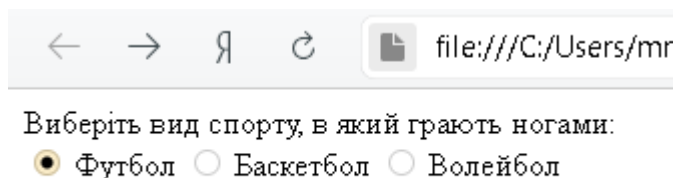


Рисунок 2.46 – Вигляд в браузері перемикачів

Прапорці

<input>

Тип `type = "checkbox"` за допомогою прапорців відвідувачі сайту можуть вибрати з декількох варіантів відповіді, а також скасовувати вибір.

Атрибут `name` відправляється на сервер разом зі значенням обраних відвідувачем варіантів відповіді. В даному випадку значення атрибута `name` має бути однаковим для всіх прапорців групи [3].

Атрибут `value` вказує значення, що відправляється серверу при установці даного прапорця [3].

Атрибут `checked` вказує, що даний прапорець повинен бути встановлений за замовчуванням при завантаженні сторінки. Атрибут приймає єдине значення `checked` [3].

Приклад:

```
diplom.html
1 <html>
2 <body>
3 <form action="http://www.priklad.com/profile.php">
4 <p>Виберіть вид спорту, в який грають руками: <br />
5 <input type="checkbox" name="sport" value="foot" checked="checked" /> Футбол
6 <input type="checkbox" name="sport" value="bask" /> Баскетбол
7 <input type="checkbox" name="sport" value="voll" /> Волейбол
8 </p>
9 </form>
10 </body>
11 </html>
```

Рисунок 2.44 – Прапорці в HTML

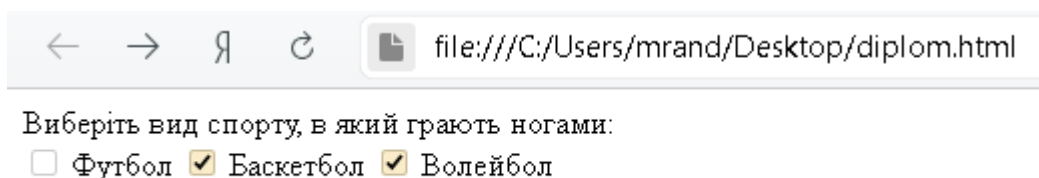


Рисунок 2.45 – Демонстрація прапорців у браузері

Також одним із основних видів форм є кнопка *підтвердження*. Без неї не може обійтись жодне меню реєстрації. Вона використовується для відправки форми з деякими даними на сервер. Для цього потрібно використати тип `type="submit"`.

Name. Даному елементу форми може бути присвоєно атрибут *name*, але його використання необов'язково [3].

Атрибут *value* використовується для вказівки тексту напису на кнопці. Встановлювати текст напису самостійно потрібно тому, що за замовчуванням в деяких браузерях на кнопці пишеться відправити (*Submit*), що підходить не для всіх форм. У різних браузерях стилі кнопок підтвердження можуть відрізнятися, найчастіше відповідаючи інтерфейсу самого браузера [3]. Приклад:

```
diplom.html x
1 <html>
2 <body>
3 <form action="http://www.priklad.com/profile.php">
4 <p>Відправити анкету <br />
5 <input type="text" name="email" />
6 <input type="submit" name="vidpraviti" value="Відправити" />
7 </p>
8 </form>
9 </body>
10 </html>
```

Рисунок 2.46 – Кнопка підтвердження (код)

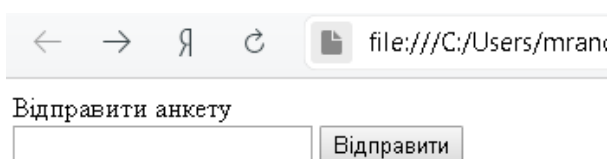


Рисунок 2.47 – Кнопка підтвердження (браузер)

2.4 Використання каскадних таблиць стилів (CSS)

У цьому розділі розглянемо, як зробити веб-сторінки більш привабливими, керуючи їх дизайном за допомогою CSS.

Каскадні таблиці стилів дозволяють створювати правила, які визначають зовнішній вигляд контенту. Наприклад, можна зробити так, щоб фон сторінки став кремового відтінку, а текст абзаців був відформатований сірим кольором і шрифтом Arial, або, щоб всі заголовки першого рівня відображалися шрифтом Times, синім кольором і курсивом [10, с. 113].

Уявімо, що кожен елемент сайту знаходиться у невидимому блоці, який керується за допомогою CSS стилів.

Існують 2 типи елементів: вбудовані та блочні. До вбудованих належать такі, як ``, `<i>`, ``, `` і ``. До блочних - `<h1>` - `<h6>`, `<p>`, `<div>`.

2.4.1. Приклади стилів:

Блоки. Ширина і висота; межі (колір, товщина і стиль); колір фону і фонові зображення; розташування у вікні браузера [12].

Текст. Шрифт: розмір, колір, курсив, напівжирне накреслення, великі літери, малі літери, капітель.

Треба запам'ятати, що таблиці стилів CSS зв'язують правила стилів HTML елементів. Ці правила вказують на те, як повинен відобразитися елемент. Правило CSS складається з двох частин: селектор та позначення його [12].

Наприклад:

p

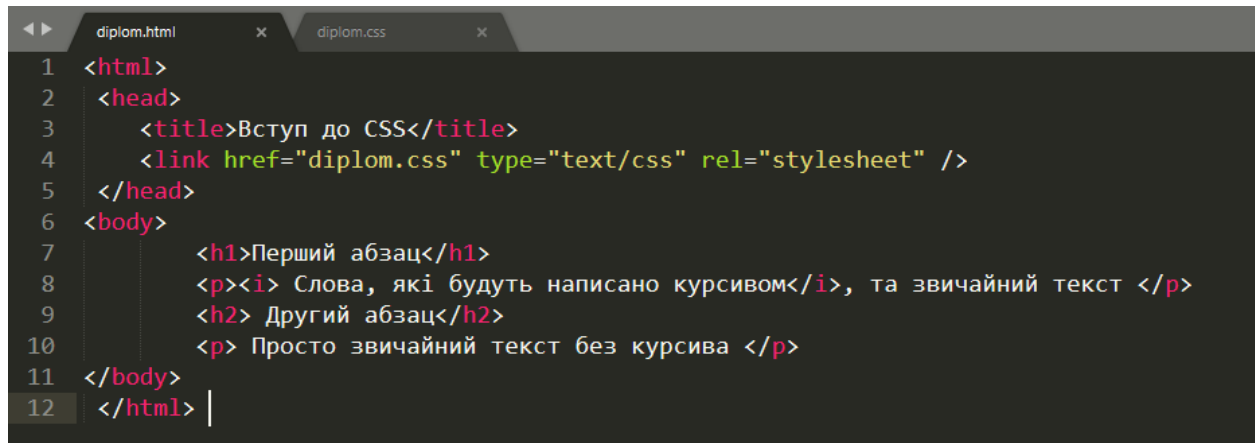
```
{ font-family: Times }
```

Селектором вважається елемент “p”. Дане правило вказує на те, що всі елементи “p” повинні бути виправлені під шрифт Times. Саме селектор вказує на те, до якого елемента буде відноситися дане правило. Позначення селектора показує, як саме буде відобразитися даний елемент. Вони знаходяться в фігурних дужках та мають властивості та значення властивостей [14].

Наприклад: h1

```
{ font-family: Times  
  
color: red      }
```

Властивості вказують на аспекти елемента, які ви бажаєте змінити, а значення – це параметр, який бажаєте використати для вибраних властивостей. Приклад:



```
1 <html>  
2 <head>  
3   <title>Вступ до CSS</title>  
4   <link href="diplom.css" type="text/css" rel="stylesheet" />  
5 </head>  
6 <body>  
7   <h1>Перший абзац</h1>  
8   <p><i> Слова, які будуть написано курсивом</i>, та звичайний текст </p>  
9   <h2> Другий абзац</h2>  
10  <p> Просто звичайний текст без курсива </p>  
11 </body>  
12 </html> |
```

Рисунок 2.48 – HTML код стилів



```
1 body  
2 { font-family: Arial, Verdana, sans-serif;}  
3 h1,h2 |  
4 { color: #ee3e80;}  
5 p  
6 {color: #665544;}
```

Рисунок 2.49 – CSS код стилів

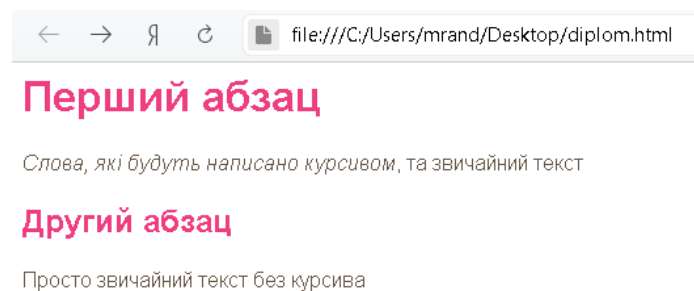


Рисунок 2.50 – Вигляд стилів у браузері

Можна зробити висновок, що за допомогою стилів CSS текст набрав більш зручнішого та привабливішого кольору.

Далі розглянемо, як підключити CSS файли до HTML [7].

Елемент `<link>` в HTML-документі повідомляє браузеру, де знаходиться файл CSS, який використовується для форматування сторінки. Це порожній елемент (він не потребує наявності закриваючого тега), який розташовується всередині елемента `<head>`.

Елемент `<link>` повинен використовувати три наступні атрибути [7].

href вказує шлях до файлу CSS, який часто знаходиться в папці з ім'ям *css* або *styles*.

type даний атрибут визначає тип документа, на який вказує посилання. Має бути вказано значення *text / css*.

rel визначає відношення між HTML-сторінкою і файлом. При створенні посилання на файл CSS має бути вказано значення *stylesheet*.

2.4.2. Кольори та їх значення.

Основний колір (color). Властивість *color* дозволяє вказати колір тексту всередині елемента. У CSS можна задати будь-який колір одним з трьох способів. Значення RGB – ці значення виражають колір в пропорціях червоного, зеленого і синього по моделі RGB, наприклад `rgb(100,100, 90)`. Шістнадцятирічні коди це шестизначні коди, що визначають кількість червоного, зеленого і синього кольору, в записі їм передують символ решітки (#), наприклад `#ee3e80`. Існує 147 імен кольорів, які розпізнаються браузерами, наприклад `DarkCyan` або `Red`.

Колір фону (background-color). Каскадна таблиця стилів звертається з кожним html елементом так, ніби він укладений в блок, а властивість *background-color* задає колір фону цього блоку. Можна вказати колір фону тими ж трьома способами, що і для задання *основного* кольору, використовуючи значення RGB, шістнадцятирічні коди і

імена кольорів. Якщо не вказати колір фону, то він буде прозорим. За замовчуванням вікна більшості браузерів мають білий фон, проте користувачі можуть самостійно задавати колір фону в налаштуваннях браузера. Тому, якщо бажаєте отримати потрібний вам колір, застосуйте властивість *background-color* до елемента `<body>` [12].

Кожен відтінок на екрані комп'ютера створюється з-за допомогою змішування червоного, синього та зеленого кольорів в різних пропорціях. Саме позначення RGB вказує на кольори Red, Green, Blue.

2.4.3. Текст у CSS. Вказівка шрифту (*font-family*).

Властивість *font-family* дозволяє вказати гарнітуру шрифту для тексту всередині елементів, до яких застосовується правило CSS. Значенням цієї властивості є ім'я гарнітури, яку ви бажаєте використовувати. Щоб сайт коректно відображався, на комп'ютерах відвідувачів повинна бути встановлена зазначена гарнітура шрифту. Можна вказати список шрифтів, розділених комами, щоб у разі відсутності першої гарнітури браузер використовував альтернативний шрифт зі списку. Крім того, частою практикою є вказівка в кінці списку загальної назви типу шрифтів. Якщо ім'я шрифту складається більш ніж з одного слова, то його потрібно вказувати в подвійних лапках. Дизайнери вважають, що сторінки зазвичай виглядають краще, якщо на них використовується не більше трьох шрифтів [12].

Розмір шрифту (font-size). Властивість *font-size* дозволяє вказати розмір шрифту. Існує кілька способів зробити це. Найбільш поширені наступні:

Пікселі. Значення в пікселях використовуються часто, оскільки такий підхід надає веб-дизайнерам дуже точний контроль над тим, скільки місця буде займати їх текст. Числове значення супроводжується буквами *px*.

Відсотки. За замовчуванням розмір тексту в браузерах дорівнює 16px. Таким чином, значення 75% відповідає 12px, а 200 % - 32px. Якщо ви задасте правило, що всьому тексту всередині елемента `<body>` значення 75% від розміру (що відповідає

12px), а потім створите ще одне правило, яке задає значення 75% вмісту елемента, вкладеного в `<body>`, то розмір шрифту в цьому елементі буде дорівнювати 9px (75% від 12px) [14].

Одиниці *Em* – це одиниця виміру, еквівалентна ширині літери *m*.

Додаткові шрифти @Font-face.

Властивість `@font-face` дозволяє використовувати шрифт, навіть якщо він не встановлений на комп'ютері користувача, надаючи шлях до файлу шрифту, який в цьому випадку буде завантажений. Оскільки даний метод дозволяє завантажувати файл шрифту на комп'ютер користувача, дуже важливо, щоб відповідна ліцензія передбачала подібне використання. Шрифт додається до таблиці стилів за допомогою правила `@font-face`, як показано в прикладі [14].

`font-family` дана властивість дозволяє вказати ім'я шрифту. Потім воно буде використане в якості значення властивості `font-family` в іншій частині таблиці стилів (як показано в правилі для елементів `<h1>` і `<h2>`).

`src` – цей атрибут визначає шлях до файлу шрифту. Щоб прийом працював у всіх браузерах, необхідно вказати шлях до декількох різних версій шрифту.

Напівжирне накреслення (`font-weight`). Властивість `font-weight` дозволяє формувати шрифт тексту напівжирним накресленням. Зазвичай дана властивість має два значення:

`normal` – використання даного значення призводить до того, що текст має звичайну вигляд.

`bold` – це значення відповідає напівжирному накресленню. В даному прикладі показано, що елемент, чий атрибут `class` має значення `credits`, виділений шрифтом з напівжирним накресленням.

Ймовірно, цікаво, навіщо передбачається `normal` шрифт. Якщо, наприклад, ви створили правило для елемента `<body>`, що вказує, що весь текст в тілі документа

повинен бути виділений напівжирним, вам може знадобитися виділити текст в окремих областях звичайним шрифтом. Даний метод в основному використовується в якості перемикача [12].

2.4.4. Використання блоків.

width, height

За замовчуванням розмір блоку задається так, щоб він був здатний вмістити контент. Щоб встановити власні розміри блоку, можна використати властивості *height* і *width*. Найчастіше розмір блоку вказується в пікселях, відсотках або одиницях *em*. Пікселі найбільш популярні, оскільки дозволяють більш точно контролювати розмір. При використанні відсотків розмір блоку задається щодо розміру вікна браузера, а якщо блок укладений в інший блок, то щодо його розміру. При використанні одиниць *em* розмір блоку залежить від розміру тексту, що міститься в ньому. Останнім часом розробники почали частіше використовувати відсотки і одиниці *em* при створенні гнучких дизайнів, що передбачають доступ з пристроїв з різними розмірами екранів [7].

min-height, max-height

Так само як і у випадку з шириною блоку, можна обмежувати його висоту. Це досягається за допомогою властивостей *min-height* і *max-height*. Якщо блок недостатньо великий, щоб вмістити контент, той виходить за його межі, що виглядає дуже негарно. Щоб вирішити проблему, що виникає при нестачі місця в блоці, можна використовувати властивість *overflow*.

Якщо брати до уваги весь блок, то він має границі, поля та відступи. За допомогою цих характеристик можна управляти зовнішнім видом блока.

Необхідно знати, що кожен блок має границі, якщо вони мають товщину 0 пікселів, то вони все рівно існують. Вони відділяють початок і кінець двох різних блоків. Кожен блок має поля, які знаходяться за межами границь. Можна вручну налаштувати ширину поля, щоб налаштувати інтервал між границями блоків.

Також треба пам'ятати і про відступи. Вони потрібні для того, щоб створити інтервал між границями блока і його вмістом [7].

Можна налаштовувати стиль кордонів, використовуючи властивість *border-style*. Вона здатна приймати такі значення:

Solid – безперервна лінія;

Dotted – пунктирна у вигляді точок (якщо товщина кордону дорівнює 2 пікселям, то розмір точки становить 2 x 2 пікселя, а інтервал між точками - 2 пікселя);

Dashed – пунктирна;

Double – дві суцільні лінії (значення властивості *border-width* створює поєднання двох ліній);

Groove – лінія, яка надає вигляд врізаної кнопки;

Ridge – опукла лінія;

Inset – вигляд вдавненої лінії;

Outset – лінія, з якої утворюється видавлена кнопка ;

Hidden / none – кордону не видно.

Можна змінювати стиль окремих кордонів, використовуючи наступні властивості:

Border-top-style;

Border-left-style;

Border-right-style;

Border-bottom-style.

Border-color

Можна вказати колір кордонів, використовуючи значення RGB, шістнадцятирічні коди або Імена кольорів. Кольорами окремих меж блоку можна керувати за допомогою наступних властивостей:

Border-top-color

Border-right-color

Border-bottom-color

Border-left-color.

Крім того, можна вказати різні кольори для всіх чотирьох кордонів в одній властивості.

Border-color: darkcyan, deeppink darkcyan, deeppink.

Тут значення перераховані за годинниковою стрілкою: верхня, права, нижня і ліва межі. Також можна використовуючи значення HSL вказати колір. Однак цей метод був введений тільки в CSS3 і не спрацьовує в старих версіях браузерів [14].

Відступи. Властивість *padding* дозволяє вказати, скільки місця має бути між вмістом елемента і його межами. Значення цієї властивості найчастіше вказується в пікселях (хоча також можна використовувати відсотки або одиниці *em*). При використанні процентних значень розмір відступу розраховується в процентному відношенні до розміру вікна браузера (або навколишнього блоку, якщо він є). Якщо для блоку вказана ширина, то значення відступу додається до значення ширини. Можна задати різні значення для кожної сторони блоку, використовуючи властивості:

Padding-top;

Padding-right;

Padding-bottom;

Padding-left.

Також можна використовувати стенографічну властивість, де значення перераховуються за годинниковою стрілкою (верх, права сторона, низ, ліва сторона: padding: 10px, 5px, 3px, 1px) [14].

2.4.5. Форми, таблиці та списки.

Позиціонування маркера.

List-style-position. Для списків за замовчуванням створюються відступи, а властивість *list-style-position* вказує, чи повинен маркер розташовуватися всередині або зовні блоку, що містить елементи списку. Дана властивість може приймати одне з двох значень [13].

Outside. При цьому маркер розташовується зліва від текстового блоку (поведінка за замовчуванням, якщо дана властивість не використовується).

Inside. Маркер розташовується всередині блоку з текстом, для якого створений відступ.

Властивості таблиць

Width - встановлює ширину таблиці.

Padding - визначає відстань між межею кожної комірки і її вмістом.

Text-transform - використовується для перемикання регістра заголовків таблиці на верхній.

Letter-spacing, font-size - додаються додаткові стилі до заголовків таблиці.

Border-top, border-bottom - додаються межі над і під заголовками таблиці.

Text-align - вирівнюється текст в осередках по лівому і по правому краю.

Background-color - замінюється фоновий колір рядків таблиці.

Hover - виділяється рядок таблиці при наведенні на неї покажчика миші.

Щоб таблиця краще сприймалась слід звернути уваги на деякі речі.

Необхідно створювати відступи в комірках, якщо текст в комірках таблиці торкається меж (або іншої комірки), то таблицю набагато важче сприймати. Додавання відступу покращує читабельність [13].

Треба виділяти заголовки напівжирним накресленням шрифту (стиль за замовчуванням для елементів <th>). Це також покращує читабельність. Також можна використовувати верхній регістр для тексту заголовків і додавати фоновий колір або підкреслення [13].

Необхідно відтіняти рядки, що допомагає користувачам слідувати по рядках. Треба використовувати невелику відмінність від звичайного кольору рядків таблиці, щоб вона виглядала охайно [13].

Вирівняти цифри можна використовуючи властивість *text-align*. Вирівняти вміст кожного стовпця, що містить цифри, по правому краю так, щоб великі цифри відрізнялися від більш дрібних.

Границі *пустих* комірок. Якщо у вашій таблиці є порожні клітинки, то ви можете використовувати властивість *empty-cells*, щоб вказати, чи слід відображати їх межі. Оскільки браузері по-різному поведуться з порожніми осередками, якщо ви хочете явно вказати, що межі повинні бути показані або приховані, то вам слід використовувати дану властивість. Вона може приймати одне з трьох значень [7].

Show. При використанні цього значення межі порожніх осередків відображаються.

Hide. При використанні цього значення межі порожніх осередків ховаються.

Inherit. Якщо одна таблиця вкладена в іншу, то дане значення призводить до того, що до осередків вкладеної таблиці застосовуються правила, що має ця таблиця.

Приклад:


```
1 <html>
2 <head>
3 <title>Вступ до CSS</title>
4 <link href="diplom.css" type="text/css" rel="stylesheet" />
5 </head>
6 <body>
7 <table class="one">
8 <tr>
9 <td>1</td>
10 <td>2</td>
11 </tr>
12 <tr>
13 <td>3</td>
14 <td></td>
15 </tr>
16 </table>
17 </body>
18 </html>
```

Рисунок 2.51 – Границі пустих комірок (код)

```
1 td
2 { border: 1px solid #0088dd; padding: 15px;}
3 table.one
4 { empty-cells: show; }
5 table.two
6 { empty-cells: hide; } |
```

Рисунок 2.52 – CSS границі комірок

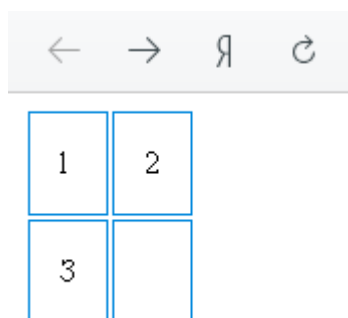


Рисунок 2.53 – Границі комірок (браузер)

Інтервали між комірками. Властивість *border-spacing* дозволяє налаштувати відстань між сусідніми комірками. За замовчуванням браузери зазвичай залишають невеликий інтервал між комірками таблиці, і, щоб збільшити або зменшити його, можна використовувати властивість. Значення цієї властивості зазвичай задається в пікселях. Можна вказувати два значення, якщо бажаєте окремо налаштувати горизонтальні і вертикальні інтервали [7].

Якщо для комірок таблиці задана межа, то в місцях з'єднання комірок лінія буде в два рази товща, ніж по краях таблиці. Щоб запобігти цьому, слід використовувати властивість *border-collapse*. Можливі значення її перераховані нижче [7].

Collapse. При використанні даного значення межі зливаються в одну (комірки будуть здвинуті, а властивості *border-spacing* і *empty-cells* - проігноровані).

Separate. Границі відокремлюються одна від одної. (Властивості *border-spacing* та *empty-cells* виконуються). Приклад:

```
diplom.html x diplom.css x
1 <html>
2 <head>
3   <title>Вступ до CSS</title>
4   <link href="diplom.css" type="text/css" rel="stylesheet" />
5 </head>
6 <body>
7 <table class="one">
8   <tr>
9     <td>1</td>
10    <td>2</td>
11  </tr>
12  <tr>
13    <td>3</td>
14    <td></td>
15  </tr>
16 </table>
17 </body>
18 </html>
```

Рисунок 2.54 – Межі границь HTML

```
diplom.html x diplom.css x
1 td
2 { background-color: #0088dd;
3   padding: 15px;
4   border: 2px solid #000000;}
5 table.one
6 { border-spacing: 5px 15px;}
7 table.two
8 { border-collapse: collapse;}
```

Рисунок 2.54 – Межі границь CSS

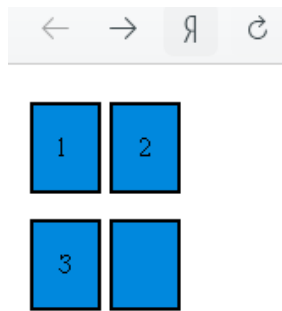


Рисунок 2.55 – Вигляд границь у браузері

2.5 Об'єктна модель JavaScript

Для того, щоб розібратися в JavaScript потрібно розуміти, що це не Java. Це дві абсолютно різні мови програмування. JavaScript не має нічого спільного з мовою Java, крім схожого синтаксису. Головне, JavaScript може звертатися до зовнішніх властивостей і методів, які вбудовані в HTML-сторінку [2].

Сценарії JavaScript розміщуються всередині web-сторінки і не можуть існувати окремо від неї. Для виконання JS-сценаріїв не потрібен компілятор, вони виконуються браузером. JS-сценарій – це звичайний текст, і можна переглянути код сценарію неозброєним поглядом – без будь-якого спеціального програмного забезпечення [3, с. 13].

Що таке JavaScript? Це мова програмування з синтаксисом, схожим з мовою Java, що використовується в складі HTML-сторінок для збільшення їх функціональності. Спочатку мова JavaScript була розроблена компаніями Netscape і Sun Microsystems, де за основу було взято мову Sun Java. JavaScript дозволяє реалізовувати ті функції сторінки, які неможливо реалізувати стандартними тегамі HTML. Сценарії запускаються в результаті настання якої-небудь події, наприклад користувач натиснув кнопку або змінив розмір вікна. JavaScript має доступ до властивостей

документа та властивостей браузера. Наприклад, на JavaScript можна легко змінювати заголовок вікна браузера або текст в рядку стану[3, с. 14].

Для написання сценаріїв JavaScript не потрібно ніяке спеціальне програмне забезпечення – досить простого текстового редактора. А отже, можна використовувати HTML-редактор для написання JavaScript-коду.

У JavaScript використовується об'єктна модель документа, в рамках якої кожен HTML-контейнер можна розглядати як сукупність властивостей, методів і подій, що відбуваються в браузері. По суті, це зв'язок між HTML-сторінкою і браузером [2].

Існує старший клас Window, що дозволяє звертатися до методів і властивостей HTML-сторінки і браузера. Наприклад, метод *close* () дозволяє закрити вікно браузера, а властивість *location* – звернутися до адресного рядка браузера.

Якщо користувач не знайомий з об'єктно-орієнтованим програмуванням, тоді, напевно, він не особливо розумітиме, про що йде мова, коли говориться про об'єкти і методи. У JavaScript все базується на класах та об'єктах (оскільки це об'єктно-орієнтована мова), і без них неможливо написати свої програми. Саме тому необхідно вивчати об'єктну модель JavaScript [2, с. 23].

Об'єкт можна сприймати як сукупність даних і методів (функцій) для їх обробки. У JavaScript з деякими об'єктами також зв'язуються певні події.

Отже, що таке об'єкт? Уявімо, що об'єкт – це людина. Допустимо, наш об'єкт називається *Human*. У такого об'єкта може бути маса характеристик – ім'я, стать, дата народження, тощо. Все це називається властивостями об'єкта. З об'єктом може бути пов'язана яка-небудь подія. Наприклад, при народженні людини може генеруватися подія *Birth*. Для кожної події можна визначити її обробник-функцію, яка буде на неї реагувати. Що буде робити ця функція, залежить від події. Наприклад, обробник *Birth* може заносити в якусь загальну таблицю бази даних інформацію про об'єкт – ім'я, стать і дату народження. Така таблиця може використовуватися для прискорення пошуку потрібного об'єкта [3].

Необхідно розглянути ще одне, більш абстрактне, поняття – клас. Клас – це зразок об'єкта, можна вважати – тип змінної об'єкта. Нехай ми розробили клас *Human*, тоді об'єкт, тобто екземпляр класу *Human*, може називатися *Human1*. Можна створювати кілька об'єктів класу *Human* – імена у них будуть різними [2].

3 Практична реалізація

У даній роботі реалізований сайт магазину з назвою “GuitarLand”. Він містить в собі розділи та перехід на інші сторінки, меню реєстрації, інформацію про магазин, тощо...

Сайт був розроблений на основі HTML та CSS. Була додана доповнена відеотека та аудіотека.

Головна сторінка має такий вигляд:

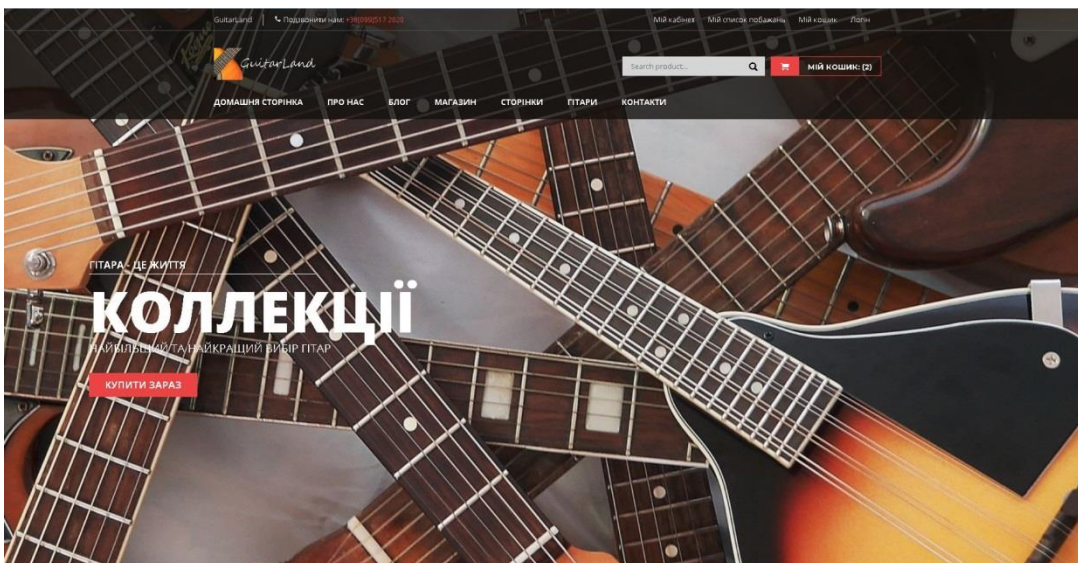


Рисунок 3.1 – Вигляд головної сторінки сайту

Проектом передбачене випадаюче меню, що складається зі сторінок сайту, на які можна перейти.



Рисунок 3.2 – Вигляд випадного меню сайту

Головна сторінка містить в собі інформацію про всі інші розділи та найкращі продукти.

Наприклад:

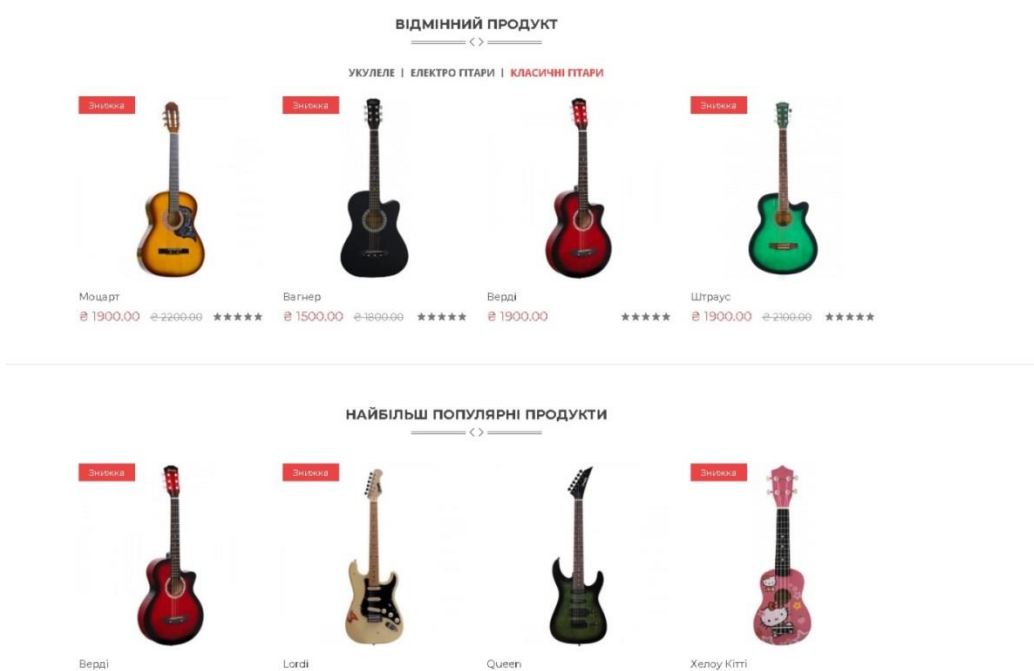


Рисунок 3.3 – Розділи продукції

Погортавши вниз, можна побачити відмінні продукти з кожного розділу сайту. Також є вибір найбільш популярних продуктів. Якщо на деякі з них є знижка, вона позначена червоною кнопкою з білим надписом. Кожна колекція гортається в лівий

та правий бік. Є в наявності ексклюзивна колекція, яка має в собі всі типи продуктів, які мають найбільший попит у клієнтів.

ЕКСКЛЮЗИВНА КОЛЕКЦІЯ

< >




 <p>Верді ★★★★★ €1900.00 €2200.00</p>	 <p>Queen ★★★★★ €4100.00 €4300.00</p>	 <p>Kiss ★★★★★ €3900.00 €4400.00</p>	 <p>Scorpions ★★★★★ £4300.00 £4800.00</p>
 <p>Lordi ★★★★★ €4000.00 €4300.00</p>	 <p>Хеллоу Кітті ★★★★★ €900.00</p>	 <p>Nirvana ★★★★★ €3500.00 €4100.00</p>	 <p>Веселий Губка Боб ★★★★★ €1100.00</p>

Рисунок 3.4 – Ексклюзивна колекція

Магазин веде свій власний блог, в якому публікує цікаві статті. Тут продаються продукти таких брендів, як вказані нижче.

ОСТАННІ НОВИНИ З БЛОГУ

< >

 <p>ЗА ВІЛЬНИЙ ЧАС in Без рубрики 15 травень 2020 У вільний час можете приділити своїй гітарі, навчити грати свого родича або друга...</p>	 <p>ЗА ВІЛЬНИЙ ЧАС in Без рубрики 16 травень 2020 Навчивши свого друга грати на гітарі - спробуйте зіграти разом...</p>	 <p>ЗА ВІЛЬНИЙ ЧАС in Без категорії 17 травень 2020 Якщо у вас вийшло зіграти разом, може спробуєте знайти ще більше друзів?</p>
---	--	--

НАШИ БРЕНДИ

< >

					
---	---	---	---	--	---

Рисунок 3.5 – Новини та бренди

Сам магазин містить в собі 3 види гітар: укулеле, електро та класичні. На кожен вид продукту надається ціна та знижка (якщо вона є). Декілька видів фото з різних боків, допоможуть вибрати саме той продукт, який ви бажаєте. Початок сторінки виглядає так:

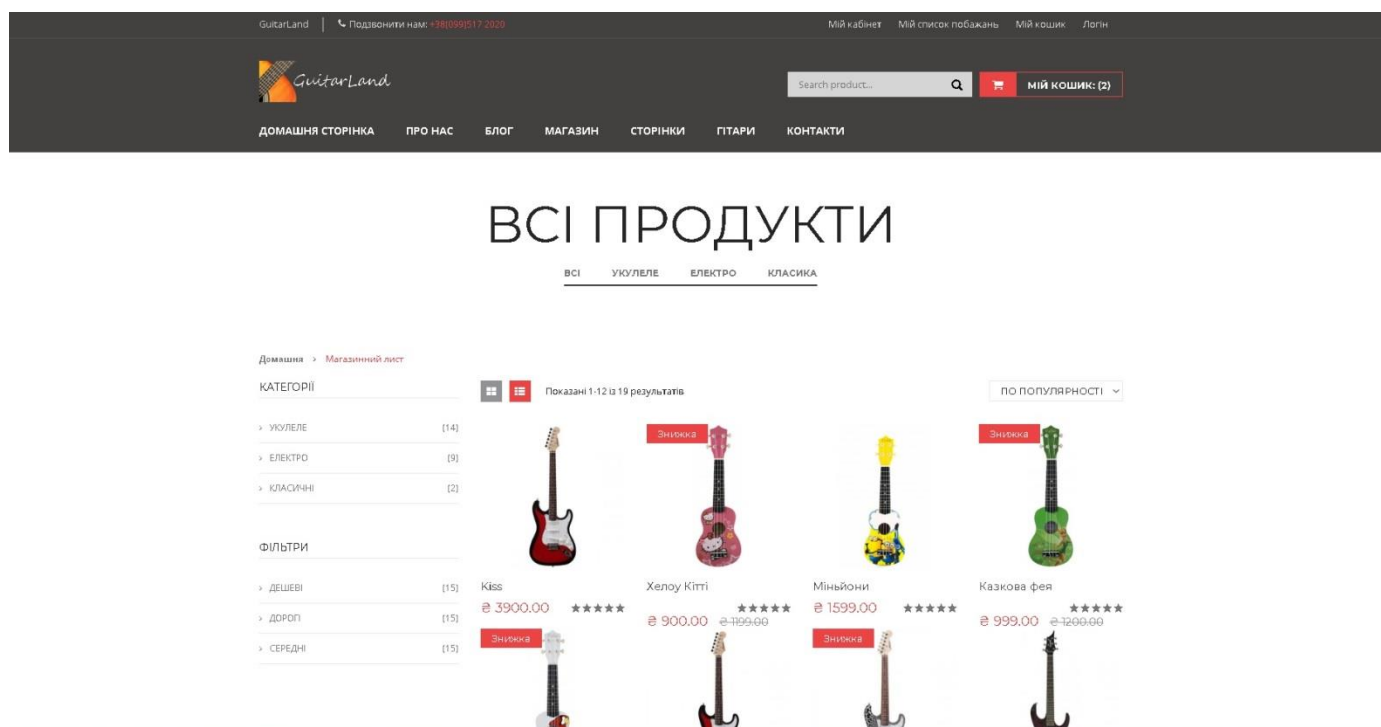


Рисунок 3.6 – Сторінка продуктів з початку

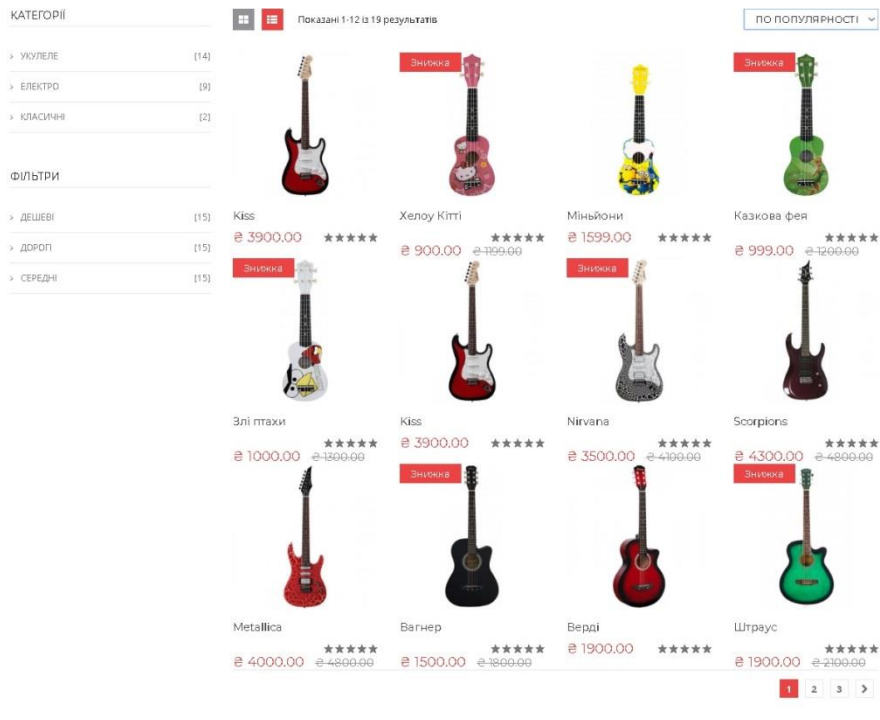


Рисунок 3.7 – Вид на сторінку продуктів з кінця

Якщо натиснути на продукт, то відкриється нова сторінка з повними даними про гітару, а саме: невеликий, але точний опис продукту (знижка, якщо присутня), ціна, доступність на складі (є, або нема), кількість для замовлення, можливість додання до кошика. Кожен може залишити свій коментар або відклик на продукт.

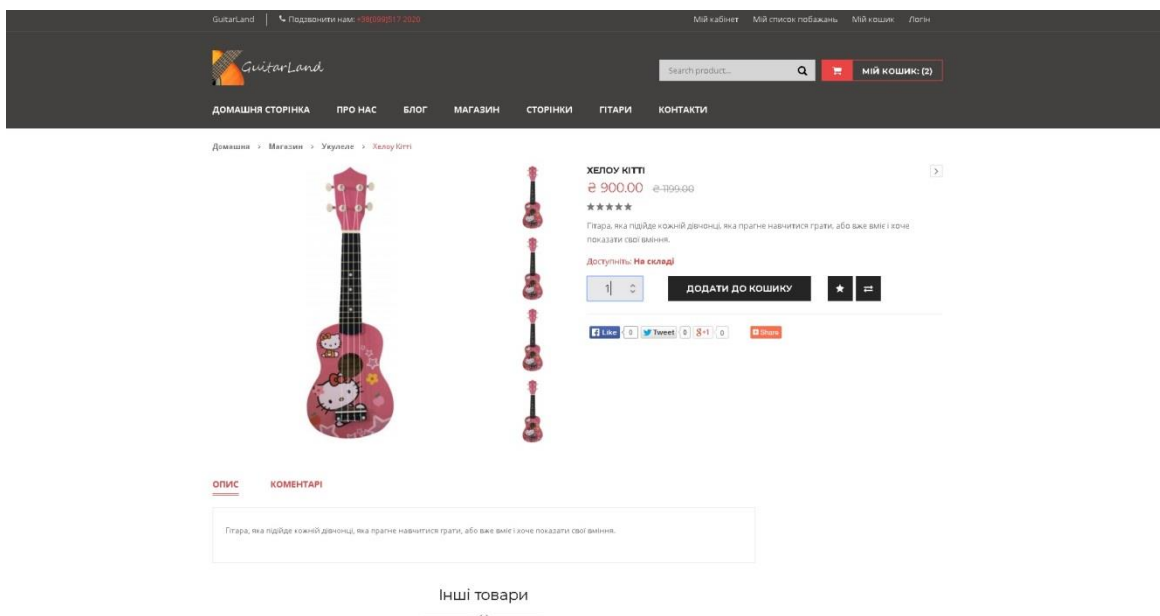




Рисунок 3.8 – Сторінка продукту

Якщо вибрана гітара задовольняє покупця, то він може додати до кошику і у нього відкриється нова сторінка. Замовник може вибрати не один товар, а декілька. Також надається можливість вибрати доставку безкоштовну (довгу), або платну (швидку). Після всіх вибраних товарів можна ввести купон, який також надає знижку. Із самого низу сторінки є підсумок в якому зазначена сума до оплати.

КАРТКА

ФОТО	ПРОДУКТ	ЦІНА	КІЛЬКІСТЬ	УВЕСЬ	ВИДАЛИТИ
	Веселий Губка Баб	€ 1100.00	1	€ 1100.00	✕
	Міньйон	€ 1599.00	1	€ 1599.00	✕

КУПОН
Введіть свій код купона, якщо він у вас є.

ПІДСУМОК ПРОДУКТІВ

ПРОМІЖНИЙ ПІДСУМОК € 2699.00

ДОСТАВКА Фіксована ціна: € 320.00 Вільна доставка

Підрахунок доставки

ЗАГАЛОМ € 3319.00

Рисунок 3.9 – Замовлення в кошику

На веб-сайті присутній список побажань, який дозволяє зареєстрованим користувачам вибрати продукти, які вони бажають купити (з можливістю не додавати до кошика покупок).

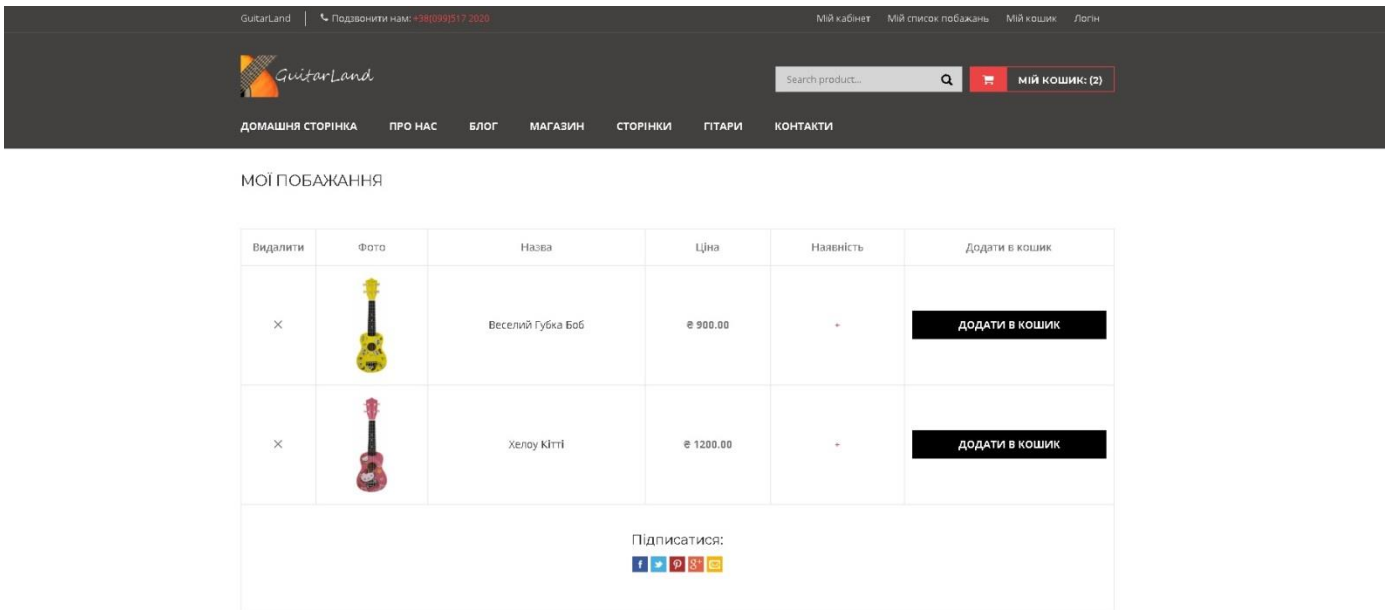


Рисунок 3.10 – Меню побажань

Кожен відвідувач має змогу зареєструватися на сайті, щоб стежити за новинами та новими гітарами. Для користувачів є меню, в якому можна авторизуватися.

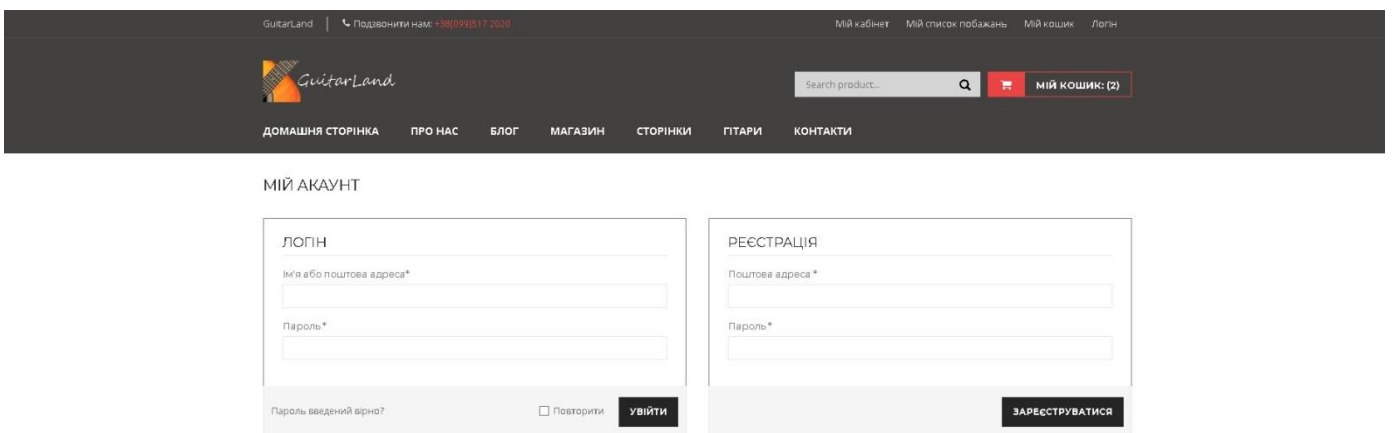


Рисунок 3.11 – Меню реєстрації та авторизації

ВИСНОВКИ

Кількість інформації у межах інтернету щорічно збільшується, тому її потрібно підносити на загальний огляд користувачів у правильному ракурсі.

Інформаційний сайт магазину відображає основну інформацію про сам магазин у зручному вигляді, розроблений з використанням сучасних технологій, має гарну структуру, відповідає всім вимогам поставленим на етапі постановки завдання.

Розробка інформаційного ресурсу була розподілена на декілька етапів:

- розробка дизайну;
- верстка;
- програмування;
- тестування;
- впровадження.

Для створення були використані:

- HTML;
- CSS;
- JavaScript;
- PHP;
- Bootstrap.

Створений продукт покращить роботу магазину, полегшить комунікацію між покупцями та продавцями, а також з адміністрацією, оскільки містить всю необхідну для цього інформацію.

Судячи з усього пройденого шляху, від дослідження торгівельної галузі до написання висновку, можна сказати лише одне, що веб-розробка – це величезна сфера комп'ютерної діяльності, яка включає в себе різні напрямки і потребує багато зусиль. Над якісним веб-сайтом повинна працювати ціла команда кваліфікованих спеціалістів.

У подальшому можна буде додавати різні функції для більш комфортнішого використання сайту. Безпосередньо, можна додати функції для людей з різними можливостями.

Виконання даної роботи дозволило закріпити вміння та знання в області веб-розробки.

СПИСОК ЛІТЕРАТУРИ

1. С. Пьюривал Основи розробки веб-додатків / С. Пьюривал // СПб.: Питер, 2015. — 272 с;
2. Кириченко А. Javascript для FrontEnd / Кириченко А. / СПб.: Наука и Техника, 2020. — 320 с;
3. Никольский А. П. JavaScript на прикладах / Никольский А. П. // СПб.: Наука и Техника, 2017. – 272с;
4. М. Кан Основи програмування на JavaScript / М. Кан // Национальный Открытый Университет “ИНТУИТ”, 2016;
5. Wallace Jackson. HTML5 Quick Markup Reference 2016 / Wallace Jackson // Library of Congress Control Number: 2016944265
6. Jennifer Niederst Robbins Learning web design a beginner’s guide to HTML, CSS, Javascript, and web graphics, Fifth Edition / Jennifer Niederst Robbins // O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
7. Ricardo Zea Mastering Responsive Web Design with HTML5 and CSS3 / Ricardo Zea // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK;
8. Сильвио Морето Bootstrap в примерах / Сильвио Морето // ДМК Пресс, 2017. – 314 с;
9. Кудрецъ Д. Практикум по HTML. / Кудрецъ Д. // Ridero, 2018;
10. С.М. Наместников Основы языка гипертекстовой разметки HTML и CSS: Учебное пособие / С.М. Наместников // УлГТУ, 2014. – с;
11. Нейт Купер Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress / Нейт Купер // Манн, Иванов и Фербер, 2019. — 266 с;
12. Леа Веру Секреты CSS. Идеальные решения ежедневных задач / Леа Веру // СПб.: Питер, 2016. — 336 с;
13. Джон Даккет HTML та CSS: Дизайн та Розробка сайтів / Дж. Дакетт // Ексмо – 480 с;

14. Eric A. Meyer and Estelle Weyl *CSS: The Definitive Guide, Fourth Edition* / Eric A. Meyer and Estelle Weyl // O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472;
15. Paul McFedries *Web Coding & Development All-in-One For Dummies* / Paul McFedries // 111 River Street, Hoboken, NJ 07030-5774;
16. Anirudh Prabhu *Beginning CSS Preprocessors* / Anirudh Prabhu // RightsLink;
17. Веру Л. *Секреты CSS. Идеальные решения ежедневных задач* / Веру Л. // СПб.: Питер, 2016. — 336 с.

ДОДАТОК

Index.html

```
<!doctype html>

<html class="no-js" lang="">

  <head>

    <meta charset="utf-8">

    <meta http-equiv="x-ua-compatible" content="ie=edge">

    <title>Домашня сторінка | Магазин гітар</title>

    <meta name="description" content="">

    <meta name="viewport" content="width=device-width, initial-scale=1">

      <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700,800' rel='stylesheet'
type='text/css'>

      <link href='http://fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>

      <link rel="shortcut icon" type="image/x-icon" href="img/logo/loggologgo.jpg">

    <link rel="stylesheet" href="css/bootstrap.min.css">

    <link rel="stylesheet" href="css/font-awesome.min.css">

    <link rel="stylesheet" href="css/owl.carousel.css">

    <link rel="stylesheet" href="css/owl.theme.css">

    <link rel="stylesheet" href="css/owl.transitions.css">

      <link rel="stylesheet" href="css/fancybox/jquery.fancybox.css">

    <link rel="stylesheet" href="css/animate.css">

    <link rel="stylesheet" href="css/meanmenu.min.css" media="all" />

    <link rel="stylesheet" href="css/normalize.css">

      <link rel="stylesheet" type="text/css" href="lib/rs-plugin/css/settings.css" media="screen" />

    <link rel="stylesheet" href="css/main.css">

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

    <link rel="stylesheet" href="css/responsive.css">

    <script src="js/vendor/modernizr-2.8.3.min.js"></script>

  </head>

  <body>

    <header class="header-home">

      <div class="header-top-area hidden-xs">

        <div class="container">
```



```

<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-7">
    <div class="header-top-left">
      <div class="top-message">GuitarLand</div>
    <div class="phone-number"> Подзвонити нам: <span>+38(099)517 2020</span></div>
  </div>
</div>
<div class="col-lg-6 col-md-6 col-sm-5">
  <div class="header-top-right">
    <div class="lang-select">
      <ul>
        <li><a href="#"></a></li>
      </ul>
    </div>
    <div class="top-menu">
      <ul>
        <li><a href="my-account.html">Мій кабінет</a></li>
        <li><a href="wishlist.html">Мій список побажань</a></li>
        <li><a href="cart.html">Мій кошик</a></li>
        <li><a href="#">Логін</a></li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
<div class="header-mid-area">
  <div class="container">
    <div class="row">
      <div class="col-lg-6 col-md-6 col-sm-4 col-xs-12">
        <div class="logo">
          <a href="index.html"></a>

```

```

        </div>
    </div>
    <div class="col-lg-6 col-md-6 col-sm-8 col-xs-12">
        <div class="cart-total">
            <ul>
                <li><a href="cart.html"><span class="cart-icon"><i class="fa fa-shopping-cart"></i></span> <span class="cart-no">Мій
                кошик: (2)</span></a>
                    <div class="mini-cart-content">
                        <div class="cart-img-details">
                            <div class="cart-img-photo">
                                <a href="#"></a>
                                <span class="quantity">1</span>
                            </div>
                            <div class="cart-img-content">
                                <a href="#"><p>Продукт 1</p></a>
                                <span></span>
                            </div>
                            <div class="pro-del"><a href="#"><i class="fa fa-times-circle"></i></a>
                            </div>
                        </div>
                    </li>
                <div class="clear"></div>
                <div class="cart-img-details">
                    <div class="cart-img-photo">
                        <a href="#"></a>
                        <span class="quantity">1</span>
                    </div>
                    <div class="cart-img-content">
                        <a href="#"><p>Продукт 2</p></a>
                        <span></span>
                    </div>
                    <div class="pro-del"><a href="#"><i class="fa fa-times-circle"></i></a>
                    </div>
                </div>
            </ul>
        </div>
    </div>

```

```

        <div class="cart-inner-bottom">
            <p class="total"><span class="amount"></span></p>
            <div class="clear"></div>
            <p class="cart-button-top"><a href="checkout.html">Перевірка</a></p>
        </div>
    </div>
</li>
</ul>
</div>
<div class="header-search">
    <form action="#">
        <input type="text" placeholder="Search product..." />
        <button type="submit"><i class="fa fa-search"></i></button>
    </form>
</div>
</div>
</div>
</div>
</div>
<div class="mainmenu-area">
    <div class="container">
        <div class="row">
            <div class="col-md-12 col-sm-12">
                <div class="mainmenu">
                    <nav>
                        <ul>
                            <li><a href="index.html">Домашня сторінка</a></li>
                            <li><a href="about-us.html">Про нас</a></li>
                            <li><a href="blog.html">Блог</a></li>
                            <li><a href="shop.html">Магазин</a>
                                <div class="mega-menu">
                                    <span>
                                        <a href="#" class="mega-title">Укулеле </a>

```

Дитячі

Для хлопчиків

Для дівчат

Електро гітари

Stratocaster

Telecaster

Les Paul

Superstrat

Класичні гітари

4 струни

5 струн

6 струн

</div>

Сторінки

<ul class="sub-menu">

Про нас

Наші контакти

Магазин

Магазинний лист

Деталі продуктів

Блог

Деталі блогу

Покупна картка

Перевірка

Лист побажань

Мій акаунт

404 сторінка

```

        </ul>

    </li>
    <li><a href="shop.html">Гітари</a></li>
    <li><a href="contact.html">Контакти</a></li>
</ul>
</nav>
</div>
</div>
</div>
</div>
</div>
<div class="mobile-menu-area">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="mobile-menu">
                    <nav id="dropdown">
                        <ul>
                            <li><a href="index.html">Домашня сторінка</a></li>
                            <li><a href="blog.html">Блог</a></li>
                            <li><a href="shop.html">Магазин</a></li>
                            <li><a href="#">Сторінки</a>
                                <ul>
                                    <li><a href="about-us.html">Про нас</a></li>
                                    <li><a href="contact.html">Контакти</a></li>
                                    <li><a href="shop.html">Сітка магазину</a></li>
                                    <li><a href="shop-list.html">Лист магазину</a></li>
                                    <li><a href="product-virtual.html">Деталі продуктів</a></li>
                                    <li><a href="blog.html">Блог</a></li>
                                    <li><a href="blog-post-img.html">Деталі блогів</a></li>
                                    <li><a href="cart.html">Магазинна картка</a></li>
                                    <li><a href="checkout.html">Перевірка</a></li>
                                    <li><a href="wishlist.html">Побажання</a></li>
                                </ul>
                            </li>
                        </ul>
                    </nav>
                </div>
            </div>
        </div>
    </div>
</div>

```



```

data-easing="easeInOutBack"
style="border-width: 2px; background-color: #ffffff; min-height:
1px;min-width: 400px;">
</div>
<div class="tp-caption large_black sfr"
data-x="150"
data-y="400"
data-speed="1100"
data-start="1400"
data-easing="easeInOutBack"
style=" font-size: 100px; font-weight: 800; line-height:80px;text-
transform: uppercase; color: #fff; text-shadow: 0 0 2px rgba(0,0,0,.05)">
Коллекції
</div>
<div class="tp-caption large_black sfr"
data-x="150"
data-y="500"
data-speed="1100"
data-start="1400"
data-easing="easeInOutBack"
style=" font-size: 20px; font-weight: normal; color: #FFF;line-
height:14px;text-transform: uppercase;">
Найбільший та найкращий вибір гітар
</div>
<div class="tp-caption lfb carousel-caption-inner"
data-x="150"
data-y="550"
data-speed="1300"
data-start="1700"
data-easing="easeInOutBack"
style=" font-size: 17px; font-weight: bold; text-transform:
uppercase; color: #F3F3F3;">
<a href="#" class="s-btn" style="background: #EC4445;color:
#ffffff;display: block;padding: 12px 28px;">купити зараз</a>
</div>
</li>

```

```

<li data-transition="random"
data-slotamount="7"
data-masterspeed="1000">
    
    <div class="tp-caption large_black sfr"
data-x="150"
data-y="350"
data-speed="1100"
data-start="1100"
data-easing="easeInOutBack"
        style="font-size: 18px; font-weight: bold; text-transform:
uppercase; color: #FFF;">
            Гітара - це життя
    </div>
    <div class="tp-caption large_black sfr"
data-x="150"
data-y="375"
data-speed="1100"
data-start="1100"
data-easing="easeInOutBack"
        style="border-width: 2px; background-color: #ffffff; min-height:
1px;min-width: 400px;">
    </div>
    <div class="tp-caption large_black sfr"
data-x="150"
data-y="400"
data-speed="1100"
data-start="1400"
data-easing="easeInOutBack"
        style=" font-size: 100px; font-weight: 800; line-height:80px;text-
transform: uppercase; color: #fff; text-shadow: 0 0 2px rgba(0,0,0,.05)">
            Колекція вишуканих гітар
    </div>
    <div class="tp-caption large_black sfr"

```



```

data-x="150"
data-y="500"
data-speed="1100"
data-start="1400"
data-easing="easeInOutBack"
style=" font-size: 20px; font-weight: normal; color: #FFF;line-
height:14px;text-transform: uppercase;">
        Найбільший та найкращий вибір гітар
</div>
<div class="tp-caption lfb carousel-caption-inner"
data-x="150"
data-y="550"
data-speed="1300"
data-start="1700"
data-easing="easeInOutBack"
style=" font-size: 17px; font-weight: bold; text-transform:
uppercase; color: #F3F3F3;">
        <a href="#" class="s-btn" style="background:#EC4445;color:
#ffffff;display: block;padding: 12px 28px;">купити зараз</a>
</div>
</li>
</ul>
</div>
</div>
<div class="banner-area">
<div class="container">
<div class="row">
<div class="col-md-5 col-sm-5 col-xs-12">
<div class="single-banner">
<a href="#"></a>
</div>
</div>
<div class="col-md-7 col-sm-7 col-xs-12">
<div class="row">

```



```

<script type="text/javascript" src="lib/rs-plugin/js/jquery.themepunch.revolution.min.js"></script>
<script src="lib/rs-plugin/rs.home.js"></script>
<script src="js/plugins.js"></script>
<script src="js/main.js"></script>
</body>
</html>

```

Повний код зазначено в додатку в окремому файлі*

About-us.html

```

<div class="col-md-6 col-sm-12 col-xs-12">
    <div class="about-text">
        <h2>GuitarLand</h2>
        <p>На сьогоднішній день інтернет-магазин GuitarLand є одним з
найпопулярніших і швидко розвиваються в Україні магазинів музичних інструментів, а також звукового і світлового
обладнання. З кожним днем нас відвідує все більше і більше клієнтів, зацікавлених в якісному обслуговуванні, швидкій
доставці товару і надійному сервісі, що ми зі свого боку гарантуємо кожному покупцеві. А наявність в м. Києві
реального магазину, який працює вже більше 10 років, дає гарантію, що в один "прекрасний день" ми не зникнемо з
Інтернету, як це часто буває.</p>
    </div>
</div>
<div class="about-counter-area">
    <div class="container">
        <div class="row">
            <div class="col-md-3 col-sm-3">
                <div class="single-counter">
                    <i class="fa fa-users"></i>
                    <h3 class="about-counter">150</h3>
                    <p>Щасливих клієнтів</p>
                </div>
            </div>
            <div class="col-md-3 col-sm-3">
                <div class="single-counter">
                    <i class="fa fa-trophy"></i>

```

```

        <h3 class="about-counter">98</h3>
        <p>Нагород</p>
    </div>
</div>
<div class="col-md-3 col-sm-3">
    <div class="single-counter">
        <i class="fa fa-thumbs-up"></i>
        <h3 class="about-counter">9999</h3>
        <p>Виповнених заказів</p>
    </div>
</div>
<div class="col-md-3 col-sm-3">
    <div class="single-counter">
        <i class="fa fa-coffee"></i>
        <h3 class="about-counter">66689</h3>
        <p>Чашок кави клієнтам</p>
    </div>
</div>
</div>
</div>
</div>
<div class="about-service-area">
    <div class="container">
        <div class="row">
            <div class="col-md-3 col-sm-4">
                <div class="single-service">
                    <h3>Головний директор</h3>
                    <div class="sercice-icon">
                        <i class="fa fa-smile-o"></i>
                    </div>
                    <div class="service-text">
                        <p>Голова магазину</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
</div>
<div class="col-md-3 col-sm-4">
  <div class="single-service">
    <h3>Заступник директора</h3>
    <div class="service-icon">
      <i class="fa fa-html5"></i>
    </div>
    <div class="service-text">
      <p></p>
    </div>
  </div>
</div>
<div class="col-md-3 col-sm-4">
  <div class="single-service">
    <h3>Продавец</h3>
    <div class="service-icon">
      <i class="fa fa-link"></i>
    </div>
    <div class="service-text">
      <p></p>
    </div>
  </div>
</div>
<div class="col-md-3 hidden-sm">
  <div class="single-service">
    <h3>Консультант</h3>
    <div class="service-icon">
      <i class="fa fa-camera"></i>
    </div>
    <div class="service-text">
      <p></p>
    </div>
  </div>
</div>
</div>
```

```

        </div>
    </div>
</div>
<!-- about-service-area end -->
<!-- about-team-area start -->
<div class="about-team-area">
    <div class="container">
        <div class="row">
            <div class="col-md-3 col-sm-4">
                <div class="single-about-team">
                    <div class="about-team-img">
                        
                    </div>
                    <div class="about-team-info">
                        <h3>Іван</h3>
                        <p>Продавець, який завжди рад продати вам товар.</p>
                    </div>
                </div>
            </div>
            <div class="col-md-3 col-sm-4">
                <div class="single-about-team">
                    <div class="about-team-img">
                        
                    </div>
                    <div class="about-team-info">
                        <h3>Анастасія</h3>
                        <p>Найкращий консультант.</p>
                    </div>
                </div>
            </div>
            <div class="col-md-3 col-sm-4">
                <div class="single-about-team">
                    <div class="about-team-img">
                        
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="about-team-info">
            <h3>Михайло</h3>
            <p>Інтернет-консультант.</p>
        </div>
    </div>
</div>
<div class="col-md-3 col-sm-3 hidden-sm">
    <div class="single-about-team">
        <div class="about-team-img">
            
        </div>
        <div class="about-team-info">
            <h3>Марина</h3>
            <p>Системний адміністратор.</p>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

```

Повний код зазначено в додатку в окремому файлі*

My-account.html

```

<div class="entry-header-area">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="entry-header">
                    <h1 class="entry-title">Мій акаунт</h1>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

</div>
<div class="my-account-area">
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <form action="#">
          <div class="form-fields">
            <h2>Логін</h2>
            <p>
              <label>Ім'я або поштова адреса<span class="required">*</span></label>
                <input type="text" />
            </p>
            <p>
              <label>Пароль <span class="required">*</span></label>
                <input type="password" />
            </p>
          </div>
          <div class="form-action">
            <p class="lost_password"><a href="#">Пароль введений вірно?</a></p>
            <input type="submit" value="Увійти" />
            <label><input type="checkbox" /> Повторити </label>
          </div>
        </form>
      </div>
      <div class="col-md-6">
        <form action="#">
          <div class="form-fields">
            <h2>Реєстрація</h2>
            <p>
              <label>Поштова адреса <span class="required">*</span></label>
                <input type="text" />
            </p>
            <p>
              <label>Пароль <span class="required">*</span></label>

```



```

        <input type="password" />
    </p>
</div>
<div class="form-action">
    <input type="submit" value="Зареєструватися" />
</div>
</form>
</div>
</div>
</div>
</div>
<div class="footer">
    <div class="footer-top-area">
        <div class="container">
            <div class="row">
                <div class="col-lg-3 col-md-3 col-sm-4">
                    <div class="footer-widget">
                        
                        <p></p>
                        <div class="widget-icon">
                            <a href="#"><i class="fa fa-facebook"></i></a>
                            <a href="#"><i class="fa fa-twitter"></i></a>
                            <a href="#"><i class="fa fa-linkedin"></i></a>
                            <a href="#"><i class="fa fa-google-plus"></i></a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Повний код зазначено в додатку в окремому файлі*

Shop-list.html

```

<div class="col-lg-3 col-md-3">
    <div class="single-product">
        <span class="sale-text">Знижка</span>
        <div class="product-img">
            <a href="#">

```

```


</a>
```

```
<div class="actions">
<div class="action-buttons">
<div class="add-to-cart">
```

```
<a href="#">Додати до картки</a>
```

```
</div>
```

```
<div class="add-to-links">
```

```
<div class="add-to-wishlist">
```

```
<a href="#" data-toggle="tooltip" title="Add to Wishlist"><i class="fa fa-star"></i>
```

```
</a>
```

```
</div>
```

```
<div class="compare-button">
```

```
<a href="#" data-toggle="tooltip" title="Compare"><i class="fa fa-exchange"></i></a>
```

```
</div>
```

```
</div>
```

```
<div class="quickviewbtn">
```

```
<a href="#" data-toggle="tooltip" title="Quick View"><i class="fa fa-search-plus"></i></a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="product-content">
```

```
<h2 class="product-name"><a href="#">Хелой Китти</a></h2>
```

```
<div class="pro-rating">
```

```
<a href="#"><i class="fa fa-star"></i></a>
```

```
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
```

```
</div>
```

```
<div class="price-box">
```

```
<span class="new-price">₺ 900.00</span>
```

```
<span class="old-price">₺ 1199.00</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-lg-3 col-md-3">
```

```
<div class="single-product">
```

```
<div class="product-img">
```

```
<a href="#">
```

```

```

```

```

```
</a>
```

```
<div class="actions">
```

```
<div class="action-buttons">
```

```
<div class="add-to-cart">
```

```
<a href="#">Додати до картки</a>
```

```
</div>
```

```
<div class="add-to-links">
```

```
<div class="add-to-wishlist">
```

```
<a href="#" data-toggle="tooltip" title="Add to Wishlist"><i class="fa fa-star"></i>
```

```
</a>
```

```
</div>
```

```
<div class="compare-button">
```

```

    <a href="#" data-toggle="tooltip" title="Compare"><i class="fa fa-exchange"></i></a>

</div>

</div> <div class="quickviewbtn">

<a href="#" data-toggle="tooltip" title="Quick View"><i class="fa fa-search-plus"></i></a>

</div>

</div>

</div>

</div>

</div>

<div class="product-content">
<h2 class="product-name"><a href="#">Мінйюни</a></h2>
<div class="pro-rating">
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
<a href="#"><i class="fa fa-star"></i></a>
</div>
<div class="price-box">
<span class="new-price">€ 1599.00</span>
</div>
</div>
</div>
</div>
</div>

```

Style.css

```

html, body {height: 100%;}

.floatleft {float:left}

.floatright {float:right}

.alignleft {float:left;margin-right:15px;margin-bottom: 15px}

.alignright {float:right;margin-left:15px;margin-bottom: 15px}

.aligncenter {display:block;margin:0 auto 15px}

```

```

a:focus {outline:0px solid;text-decoration:none}
img {max-width:100%;height:auto}
.fix {overflow:hidden}
p {margin:0 0 15px;}
h1, h2, h3, h4, h5, h6 {
  color: #252525;
  font-family: 'Montserrat', sans-serif;
  line-height: 1.35;
  margin: 0 0 10px;
}
a {transition: all 0.3s ease 0s;text-decoration:none;color:#777}
a:hover {
  color: #ec4445;
  text-decoration: none;
}
a:active, a:hover {
  outline: 0 none;
}
ul{
list-style: outside none none;
margin: 0;
padding: 0
}
body {
  color: #777;
  font-family: "Open Sans",sans-serif;
  font-size: 13px;
  font-style: normal;
  font-weight: 400;
  line-height: 18px;
  visibility: visible;
}
.clear{clear:both}
.slicknav{display:none}

```

```
.container{ width:1200px }
```

Повний код зазначено в додатку в окремому файлі*