

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка інтернет-магазину підприємства торгівлі»

Студента 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

Хілько
Володимира
Володимировича

підпис студента

Науковий керівник
доктор технічних наук, професор

Краскевич Валерій
Євгенійович

підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2020

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
«20» грудня 2019р.

Завдання **на випускню кваліфікаційну роботу (проект) студенту**

Хілько Володимир Володимировичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка інтернет-магазину підприємства торгівлі»
Затверджена наказом ректора від «04» грудня 2019 р. № 4111
2. Строк здачі студентом закінченої роботи 29 травня 2020 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: створення інтернет-магазину підприємства торгівлі.
Об'єкт дослідження: процеси визначення споживчих пріоритетів у сфері торгівлі.
Предмет дослідження: сучасні інтернет технології для розробки веб-сайту
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Краскевич В.Є.	15.12.2019 р.	15.12.2019 р.
2	Краскевич В.Є.	15.12.2019 р.	15.12.2019 р.
3	Краскевич В.Є.	15.12.2019 р.	15.12.2019 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Теоретичні аспекти моделювання інтернет-магазину підприємства торгівлі

1.1. Теоретичний аналіз для визначення проекту

1.2. Поняття інформаційного наповнення сайту

1.3. Процес розробки веб-дизайну сайту

1.4. Технології написання програмного коду та верстки сайту

РОЗДІЛ 2. Огляд методик створення веб-сайтів та його складових

2.1. Використання ручного методу (HTML, CSS)

2.2. Бази даних. Їх значення та функції

2.3. Допомога Chrome DevTools у створенні веб-сайту

РОЗДІЛ 3. Створення інтернет-магазину підприємства торгівлі

3.1. Розробка веб-дизайну (макету) інтернет-магазину

3.2. Верстка, та налаштування інтернет-магазину

3.3. Тестування роботи сайту, перевірка усіх функцій

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.10.2019	01.10.2019

2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	<i>15.12.2019</i>	<i>15.12.2019</i>
3	<i>Вступ</i>	<i>03.02.2020</i>	
4	<i>РОЗДІЛ 1. Теоретичні аспекти моделювання інтернет-магазину підприємства торгівлі</i>	<i>28.02.2020</i>	
5	<i>РОЗДІЛ 2. Огляд методик створення веб-сайтів та його складових</i>	<i>06.04.2020</i>	
6	<i>РОЗДІЛ 3. створення інтернет-магазину підприємства торгівлі</i>	<i>12.05.2020</i>	
7	<i>Висновки</i>	<i>15.05.2020</i>	
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	<i>20.05.2020</i>	
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>03.06.2020</i>	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>09.06.2020</i>	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	<i>12.06.2020</i>	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «15» грудня 2019 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Краскевич В.Є.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Хілько В. В.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

30.05.2020 р.

(підпис, дата)

13. Висновок про випускню кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента _____

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми _____

(підпис, прізвище, ініціали)

Демідов П.Г.

Завідувач кафедри _____

(підпис, прізвище, ініціали)

Пурський О.І.

« _____ » 2020 р.

Анотація

У випускному кваліфікаційному проєкті здійснено комплексну розробку інтернет-магазину підприємства торгівлі. Досліджені методи створення веб-дизайну сайту. Розглянуті технології написання програмного коду та верстки сайтів. Розроблено сучасний дизайн інтернет-магазину підприємства торгівлі з використанням тенденцій останніх років. Створено інтернет-магазин підприємства торгівлі використовуючи оптимальні рішення розробки як результат порівняльного аналізу методів та технологій написання коду. Налаштовано функціонування інтернет магазину та усіх його функцій.

Ключові слова: інтернет-магазин, верстка, веб-дизайн, бази даних, макет, chrome dev tools.

Annotation

In the qualification project the complex development of the online store of the trade enterprise was carried out. Researched methods of creating a web site design. Technologies of writing program code and site layout are considered. The modern design of the online store of the trade enterprise with use of tendencies of the last years is developed. An online store of a trade enterprise was created using optimal development solutions as a result of comparative analysis of methods and technologies of code writing. The functioning of the online store and all its functions has been set up.

Keywords: online store, layout, web design, databases, layout, chrome dev tools.

Зміст

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ МОДЕЛЮВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ПІДПРИЄМСТВА ТОРГІВЛІ.....	10
1.1 Теоретичний аналіз для визначення проекту.....	10
1.2 Поняття інформаційного наповнення сайту.....	12
1.3. Процес розробки веб-дизайну сайту.....	14
1.4 Технології написання програмного коду та верстки сайту.....	16
РОЗДІЛ 2. ОГЛЯД МЕТОДИК СТВОРЕННЯ ВЕБ-САЙТІВ ТА ЙОГО СКЛАДОВИХ.....	17
2.1. Використання ручного методу (HTML, CSS).....	17
2.2. Бази даних. Їх значення та функції.....	20
2.3. Допомога Chrome DevTools у створенні веб-сайту.....	23
РОЗДІЛ 3. СТВОРЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ПІДПРИЄМСТВА ТОРГІВЛІ.....	28
3.1 Розробка веб-дизайну (макету) інтернет магазину.....	28
3.2 Верстка, та налаштування інтернет-магазину.....	32
3.3 Тестування роботи сайту, перевірка усіх функцій.....	39
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44

ВСТУП

Актуальність дослідження полягає в тому, що стрімке становлення та зростання доступності мережі інтернет, застосування найновітніших технологій комунікації у повсякденному житті людини створюють нові економічні явища. Розробка інтернет-магазину підприємства торгівлі поліпшить якість обслуговування клієнтів, так як дії обробляються в автоматичному режимі, знизить вірогідність помилки персоналу, знизить трудомісткість, та збільшить максимальну кількість заявок у день, а також це дасть можливість постійного зв'язку з клієнтами, що завжди буде актуально.

Зазвичай Інтернет магазин - це не просто сайт, в стандартному розумінні цього слова. Інтернет магазин це цілий комплекс спеціальних технічних рішень, які дозволяють власнику і адміністратору магазину швидко, просто і зручно, додавати і видаляти, виставлені на прилавок товари, редагувати їх опис, змінювати вартість товарів в будь-який час, і отримувати повідомлення про замовлення, на e-mail власника або навіть на його телефон у вигляді sms повідомлення.

Для покупців, хороший Інтернет магазин здатний надати максимально можливі зручності, у вигляді кошика, пошуку по ключовим словам, а також автоматичне підтвердження і нагадування про замовлення через кілька днів у разі несплати, на e-mail замовника.

На сьогоднішній день, ринок технічних рішень для максимально автоматизованої торгівлі через Інтернет, досить великий, і при наявності технічних знань і деякої практики в галузі електронної комерції, вибрати є з чого. Все залежить від потреб і кількості тих фінансів, які людина згодна вкласти, в створення своєї власної торгової точки в мережі Інтернет.

Метою роботи є розробка інтернет-магазину підприємства торгівлі. Для досягнення поставленої мети, треба виконати такі завдання:

- Дослідити варіанти верстки веб-сайтів
- Дослідити методи створення дизайну інтернет-магазину

- Розробити дизайн інтернет-магазину, після чого зверстати та налаштувати.

Об'єктом дослідження є процеси створення інтернет-магазину та його складових.

Предметом дослідження є методики, засоби розробки WEB-сайту, безпосередньо мова розмітки гіпертексту HTML та CSS.

Практичне значення. Матеріали дипломного проекту можуть бути використані підприємствами торгівлі для підвищення своїх прибутків шляхом збільшення клієнтів.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ МОДЕЛЮВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ПІДПРИЄМСТВА ТОРГІВЛІ

1.1 Теоретичний аналіз для визначення проекту

Створення сайту розпочинається з етапу який вміщує у собі усе, що стосується аналізу інформації та її збору. Це потрібно для початку робіт та розуміння масштабів проекту

На основі результатів опитувань, треба отримати чітке розуміння про те з якою метою, та які дії будуть виконуватися на вашому сайті та хто саме буде користуватися цим сайтом.

Спираючись на проаналізовані дані та побажання замовника розробляється технічне завдання. У технічному завданні повинні бути визначені всі характеристики: адаптація під різні монітори та девайси, технічні характеристики, структура веб-сайту, дизайн сторінок. Клієнт повинен ухвалити технічне завдання.

Після того як технічне завдання сформовано та затверджено пишеться проект сайту, визначаються типи сторінок: динамічні або статичні:

Статичні сторінки це такі сторінки, що виглядають завжди однаково, незалежно від дій користувача. Наприклад, меню організовано посиланнями на окремі сторінки, а не списком, що випадає [1].

Динамічна веб-сторінка, згенерована програмно на відміну від статичної сторінки. Сервер генерує HTML код для обробки браузером або іншим агентом користувача [2].

Типи побудови сайту:

- Сайт-візитка. Сайти цього типу мають представницькі цілі і дають можливість розмістити коротку інформацію про вашу фірму, сфери її діяльності та послуги, що надаються. Як правило, це 5-10 сторінок: про компанію, послуги, прайс-лист, контакти, план проїзду тощо;

•Корпоративний сайт - це представництво вашої компанії в мережі Інтернет. На відміну від сайту-візитки він містить розширений обсяг інформації. Сайт допоможе розповісти потенційним замовникам про існування вашої фірми, напрямки її діяльності, про товари і послуги що надаються. Тут можна ознайомитися з повним переліком продукції підприємства і вивчити прайс-лист. Основними завданнями корпоративного сайту є просування на ринок товарів, пошук нових замовників, покупців, партнерів і постачальників. Наявність корпоративного сайту - це один із шляхів формування позитивного іміджу організації та зміцнення довіри до неї, як до надійного партнера. Це необхідний інструмент успішного бізнесу у наш час;

•Промосайт – основна мета промо-сайтів – просування на ринок торговельної марки, товарів або послуг. Промо-сайти містять інформацію про рекламувану фірму чи продукцію, великий акцент зроблено на вигідне графічне представлення товару, використання відео або анімаційних роликів, проведення опитувань, розіграші призів. Як правило, такі сайти дуже яскраві і динамічні, вони нагадують рекламні ролики.

•Інформаційний портал - самий насичений з сайтів по структурі і набору функцій. Він призначений для управління великими обсягами інформації. Інтернет-портал - це великий сайт з певної тематики, що охоплює всі її аспекти, здатний задовольнити будь-які запити користувача. Наприклад, автомобільний портал інтегрує в собі все, пов'язане з автомобілями: автомобільні новини країни та світу, покупка, продаж, ремонт автомобілів, цікаві статті та корисна інформація, реклама, форум, на якому автолюбители можуть поспілкуватися між собою та інше. Особливістю інформаційного порталу є велика кількість відвідувачів. У створенні такого проекту зацікавлені компанії, що надають послуги, пов'язані зі значними інформаційними потоками (різні ЗМІ, контент-проекти і т.д.);

- Каталог (сайт-вітрина) містить інформацію про фірму, а також каталоги товарів з їх характеристиками, докладним описом, малюнками, фотографіями і т.д;
- Інтернет-магазин - електронні магазини багато в чому схожі на каталоги (сайти-вітрини). Основна мета - продаж товарів через Інтернет. Тут передбачається система реєстрації користувачів, можливість оформити замовлення на товар. Замовлення по e-mail передається менеджеру з продажу [3].

1.2 Поняття інформаційного наповнення сайту

Адекватне інформаційне наповнення для більшості сайтів є необхідним та фундаментальним. Виключення складають ряд Інтернет-сервісів типу безкоштовної пошти.

Під час розробки концепції інформаційного наповнення веб-сайтів слід врахувати наступні фактори:

- Тип і формат подання інформації. Щоб визначити тип і формат подання інформації, потрібно відповісти на кілька питань. Буде достатньо використати лише звичайні тексти та графіки? Чи є необхідність інтегрувати в контент сайту відео, звукові файли, Flash-презентації та інше? Чи буде сайт складатися із статичних Html-сторінок, або користувачеві буде видаватися інформація, спеціально скомпільована для нього з базових даних (наприклад, порівняти таблиці з характеристиками продуктів і цінами на них)? У випадку динамічних сторінок з'являється можливість видати користувачеві саме ту інформацію, яка його цікавить, і саме у тому виді, у якому це буде максимально зручно для навчання, аналізу, прийняття рішень і т.д [4].
- Структурування інформації. Щоб забезпечити стабільне зростання трафіку за рахунок відвідувань цільової аудиторії, а відповідно і кількість продажів, виконується певна структуризація веб-ресурсу, адже

некоректна структура перешкоджає зручному знаходженню потрібних матеріалів відвідувачам.

Все починається з головної сторінки, яку іноді ще називають домашньою. Вона відображає основну ідею сайту, надаючи зрозумілу певну інформацію всім тим, хто зайшов на даний ресурс. Метою створення такої сторінки є позначення цілей сайту і вказівка на те, яка інформація на ньому сайті міститься. Також на головній сторінці розміщується посилання на сторінку реєстрації, якщо це необхідно, додаються великі розділи або анонси найбільш відвідуваних або основних тем ресурсу.

Підрозділи і розділи сайту з наповненням розташовуються на внутрішніх сторінках, які доступні до переходу з головної. Це і є контент, який безпосередньо впливає на зацікавленість відвідувачів в матеріалах, а також підвищує авторитетність сайту в пошукових системах. Обсяг інформації залежить від кількості таких сторінок і розміщених на них даних. Чим більший обсяг матеріалів, тим більше вага, тобто значимість сайту для ранжирування. Інформація на сторінках повинна постійно оновлюватися і поступово збільшуватися, тоді пошукові роботи реагують на це, підвищуючи позиції порталу в пошукових системах. Всі сторінки мають свої назви, а також на кожен з них є відповідна посилання. Формується меню сайту і навігація в одному стилі не більш двох рівнів. Кожна сторінка може містити пару внутрішніх посилань з коротким, але зрозумілим описом, куди вони направлять користувача. На сторінці краще уникати дублюючих посилань і зменшити кількість використання Flash і JavaScript.

Також на сторінках порталу в текстах розміщуються певні ключові словосполучення. У заголовку і тезі title бажано вжити ключовий запит.

Всі ці елементи повинні бути максимально рівномірно розподіленими по тексту [5].

- Актуалізація інформації. Відповідність пошуковому запиту (релевантність) контент повинен містити в собі відповідь на

користувача запит. Ступінь відповідності контенту визначається за різними параметрами: входженню слів, збігом словоформ, наявності синонімічних конструкцій, частоти народження слів, наявності зовнішніх посилань з шуканими словами, тематики сайту і запиту і т.д. Зручність для сприйняття людиною - ця вимога виникла в протидію спробам скласти контент сайту лише зі слів, які користувачі використовують при пошуку. Очевидно, що сайт, який містить нескладний набір ключових слів не представляє з себе ніякої цінності [6].

- Унікальність. Інформація, розміщена на сайті не повинна дублювати наявну в мережі інформацію. Дійсно корисним для користувача є сайт, який є джерелом інформації, а не копією вже наявної інформації. Більш того, копіювання інформації без дозволу автора переслідується по закону. А якщо автор дає такий дозвіл, то копіювання зазвичай супроводжується установкою посилань на джерело, що теж знижує рейтинг сайту (велика кількість вихідних посилань знижують позиції сайту у видачі) [7].

1.3. Процес розробки веб-дизайну сайту

Дизайнери часто думають про Інтернет з точки зору наповнення контентом та коду. Але кмітливі дизайнери знають, що успіх веб-дизайну не визначається кодом, інтеграцією соціальних медіа чи цікавими візуальними зображеннями.

Розробка веб-сайту, що принесе користь, вимагає продуманої стратегії в Інтернеті, орієнтованої на досягнення організаційних цілей - це може бути що завгодно: залучення відвідувачів до придбання продуктів, донесення якоїсь проблеми до громадян, представлення нового бренду відвідувачам.

Дизайнер, може стати одним з найцінніших та найвпливовіших членів веб-команди, коли він зрозуміє, як розробити онлайн-стратегію. Є багато

людей, які можуть написати код і мати думки щодо дизайну та нюансів сайту, але мало хто має масу талантів та інструментів, необхідних для створення веб-сайту, який допомагає організації досягти своїх цілей [8].

Сьогодні кожен може створити веб-сайт, але не кожен може створити їх на високому рівні. Будь-хто може створити шаблон веб-сайту за допомогою онлайн-генератора, але чи сподобається це відвідувачам і чи сприятиме це просуванню бізнесу? Створення веб-сайту - це більше, ніж просто дизайн, код та технологія. Створення веб-сайту - це як складання історії, і, як і всі унікальні історії, унікальний дизайн веб-сайту є результатом унікального процесу.

Планування. Щоб почати створювати веб-сайт спочатку потрібно чітко визначитися з побажаннями замовника або якщо дизайн робиться для вашого ресурсу потрібно чітко все розпланувати. Відвідувач хоче знати про інтернет-магазин усе, що це за компанія, персонал, яка продукція та послуги, як була створена компанія, чому вона була започаткована, і головне, що потрібно новому веб-сайту робити для просування бізнесу [9].

Створення мапи сайту. Після того як ми зібрали усю потрібну інформацію потрібно створити мапу сайту.

Мапа сайту - це, по суті, структура сторінок, які буде містити веб-сайт. Планування карти сайту до роботи над самим дизайном веб-сайту має ряд переваг, оскільки ви можете побудувати дизайн на найважливіших сторінках, більш ефективно планувати навігацію на веб-сайті та отримувати загальне уявлення про вміст, який ще потрібно написати [10].

Планування вмісту. Ця частина процесу будівництва структури веб-сайту включає поглиблене планування вмісту веб-сайту. Потрібно наповнити сторінки таким чином, щоб клієнт не пропустив основну інформацію, та був зацікавлений у подальшому перебуванні на сайті [11].

Дизайн сторінок. Цей крок процесу веб-дизайну полягає у тому, щоб спроектувати зовнішній вигляд веб-сайту. На цьому кроці створюється каркасний веб-сайт з основними елементами веб-сторінки, такими як заголовок, навігація, віджети тощо.

Завданням гарного веб-дизайну, як і всіх дизайнів, є збалансування форми та функцій. Потрібно використовувати інформацію, яку була записана на етапі планування.

Дизайн повинен мати зміст. Вміст веб-сайту - це номер один, що ви хочете показати відвідувачам. Для кожного дизайну сторінки задайте собі наступні питання:

- Яка головна мета сторінки?
- Чи зрозуміло користувачам, які дії потрібно вжити?
- Як дизайн спонукає користувачів до дій?

Усі веб-дизайнери також мають розуміти, як дизайн перетвориться на код. Якщо дизайнер не займається розробкою самостійно, то потрібно запропонувати розробнику переглянути дизайн, перш ніж здавати роботу [12].

Обов'язково підтвердьте дизайн із клієнтом, перш ніж переходити до будь-якої розробки.

1.4 Технології написання програмного коду та верстки сайту

HTML та CSS. Основою веб-документа є код, написаний на мові розмітки HTML. Мова розмітки не варто плутати з мовою програмування. В принципі, за допомогою набору команд, який пропонує для розробника сайту HTML, можна задавати всі необхідні параметри статичного веб-документа - розташування елементів (блокова розмітка), заголовки, абзаци, таблиці, зображення і т.д. А за допомогою CSS, спеціальної надбудови для HTML, можна позиціонувати всі перераховані об'єкти розмітки, змінювати їх стиль - колір, розмір, формат і т.д [13].

JavaScript. Інтерактивні та анімовані елементи, наприклад - банери, рядок, що біжить, форма зворотного зв'язку, на веб-сторінках працюють завдяки присутності сценаріїв і коду, написаного на серверних або клієнтських мовах програмування. Дуже популярні сценарії, які розробляються за допомогою мови програмування JavaScript. Такі клієнтські сценарії в своїй

роботі не використовують можливості сервера і виконуються на стороні комп'ютера користувача, тобто в браузері. Завдяки цьому додатку JavaScript відрізняються простотою і високою швидкістю роботи [14].

Серверна мова програмування. У разі, коли потрібно написання складних і об'ємних кодів, наприклад для форумів або гостьових книг, програмісти звертаються за допомогою до серверних мов програмування або програмну платформу Node.js. Коди Node.js виконуються на стороні сервера, тому їх робота може бути кілька уповільнена, оскільки залежить від швидкості з'єднання з віддаленим комп'ютером і ступеня його завантаженості. За допомогою Node.js і команд SQL (спеціальна мова запитів до реляційної бази даних) можна організувати взаємодію сайту з базами даних і створювати інтерактивні інтернет-проекти - форуми, інтернет-магазини, дошки оголошень, різні каталоги і т.д [15].

Верстка сайтів - це процес створення веб-сторінки (сторінки сайту) з використанням спеціальної розмітки і програмного коду. Для верстки використовуються такі мови програмування і розмітки, як HTML (сьогодні актуальна 5 версія), CSS3 і JavaScript.

По суті, верстка сайтів - це візуалізація графічного оформлення сайту, яку створив веб-дизайнер. Без верстальника (а саме так називається професія) неможливо застосувати намальовані дизайнером краси до движку сайту, типу WordPress, 1С-Бітрік, Джумла, MODX або будь-якого іншого. Тому веб-дизайнер і верстальник завжди працюють тандемом.

РОЗДІЛ 2.

ОГЛЯД МЕТОДИК СТВОРЕННЯ ВЕБ-САЙТІВ ТА ЙОГО СКЛАДОВИХ

2.1. Використання ручного методу (HTML, CSS)

Таблична верстка. Таблиця - це структурований набір даних, що складається з рядків і стовпців (табличні дані). Таблиця дозволяє швидко та

легко шукати значення, які вказують на певний зв'язок між різними типами даних, наприклад, про людину та її вік, або день тижня чи графік роботи місцевого басейну.

Як працює таблиця? Сенс таблиці полягає в тому, що він просто структурований. Інформація легко інтерпретується шляхом створення візуальних асоціацій між заголовками рядків та стовпців. Якщо виникає якесь питання, то відповідь можна знайти, пов'язавши відповідні заголовки рядків та стовпців.

Якщо зробити все правильно, навіть незрячі можуть інтерпретувати табличні дані в таблиці HTML - успішна таблиця HTML повинна збільшити досвід оглядових та слабозорих користувачів.

Верстка таблицями Живий приклад можна побачити на GitHub. Те, що ви помітите – це гарно читабельна таблична верстка. Це тому, що до таблиць, які можна побачити на GitHub застосовані стилі за допомогою CSS.

Щоб таблиці були ефективними в Інтернеті, потрібно надати певну інформацію про стилі за допомогою CSS, а також хорошу міцну структуру з HTML [17].

Коли НЕ слід використовувати HTML-таблиці? HTML-таблиці слід використовувати для табличних даних - саме для цього вони розроблені. На жаль, багато людей використовували HTML-таблиці для розміщення веб-сторінок, наприклад, один рядок із заголовком, один рядок із стовпцями вмісту, один рядок із колонтитулом тощо. Сьогодні набагато рідше зустрічаються сайти які звертали таблицями, але ви все ще можете їх побачити в деяких куточках Інтернету.

Коротше кажучи, використання таблиць для компоновання є поганою ідеєю. Основні причини:

Таблиці макетів зменшують доступність для людей із вадами зору: Екрани засобів, використовувані незрячими людьми, інтерпретують теги, які існують на HTML-сторінці, та читають вміст для користувача. Оскільки таблиці не є правильним інструментом для компоновання, а розмітка є більш

складною, ніж у техніках компонування CSS, вихідна інформація буде заплутаною для користувачів.

Таблиці створюють суп з тегів: Як було сказано вище, компонування таблиць, як правило, включає більш складні структури розмітки, ніж правильні методи компонування. Це може призвести до того, що код буде складніше писати, підтримувати та налагоджувати.

Коли ви використовуєте належні контейнери компонування (наприклад, `<header>`, `<section>`, `<article>` або `<div>`), їх ширина за замовчуванням становить 100% від їхнього батьківського елемента. З іншого боку, таблиці за замовчуванням приймають ширину відповідно до їх вмісту, тому необхідні додаткові заходи для створення макету таблиці для ефективної роботи на різних пристроях [18].

Блокова верстка. Тег `div` - це елемент на блоковому рівні, який являє собою розділ у веб-документі. Таким чином, `<div>` підходить для побудови структури веб-сторінок. Тег `<div>` також використовується для опису вмісту, який неможливо правильно описати іншими більш семантичними тегами. Коли розробники не розуміють смислового значення інших елементів блокового рівня, вони часто додають більше тегів `div`, ніж потрібно.

Структури, що базуються на блоках, мають більш криву навчання, ніж структури на основі таблиць. Розробник повинен знати CSS та розуміти різницю між елементами на рівні блоку та елементами вбудованого характеру, коли використовувати наслідуване позиціонування та коли використовувати абсолютне позиціонування та як вирішувати помилки браузера.

Елемент `div` візуально не такий, як елемент таблиці. Усі знають, як виглядає таблиця, але діви не такі очевидні. Гарна річ у `<div>`, що це лише один елемент. Він не загорнутий у батьківський елемент, як теги `td` у таблицях. Отже, контейнер є більш гнучким і не має обмежень його батьківського тегу [19].

Використання `div` для структури може зробити сторінку більш крихкою, коли вміст підштовхує `div` до межі. Це також може змусити стовпчики

потрапляти один під одного. Але зазвичай це стосується лише старих браузерів (зокрема IE6); новіші браузери переносять вміст у наступний стовпець.

Робота з помилками браузера спочатку може бути важкою та нудною, але з досвідом розробники можуть їх виявити та виправити набагато легше та швидше. Підтримка браузера для стандартів W3C стає все кращою та кращою. Зі зростанням популярності Firefox та Safari та впровадженням Google Chrome ми спостерігаємо велику боротьбу за частку ринку, що неминуче сприяє кращим браузерам.

Найбільша проблема тегів `div` полягає в тому, що вони використовуються занадто часто. `Div` слід використовувати лише для створення структури та як заповнювачів елементів дизайну, коли жоден інший елемент рівня блоку не може описати вміст. Тег `div` призначений для логічного групування елементів.

Глибоко вкладаючи теги `div` - це вірний шлях до пекла технічного обслуговування, і код змусить розробників задуматися двічі, перш ніж торкатися його, просто тому, що він такий нечитабельний. Правда, використання описових імен класів та структур робить код більш зрозумілим, але використовувати їх для вкладених тегів `div` не завжди просто.

Занадто багато тегів `Div` - це критерій того, що код писався не самим досвідченим спеціалістом. Блоки `<div>` використовуються, коли тематичні теги на рівні блоку не зможуть краще описати вміст; Наприклад, заголовки повинні бути загорнуті в теги від `h1` до `h5` а не в `<div>`. Написання семантичного коду зазвичай скорочує базу коду; а менше `<div>` з `float` допомагає уникнути помилок браузера [20].

2.2. Бази даних. Їх значення та функції

База даних – це будь-який збір даних або інформації, спеціально організованої для швидкого пошуку та пошуку за допомогою комп'ютера. Бази даних структуровані для полегшення зберігання, пошуку, модифікації та видалення даних у поєднанні з різними операціями з обробки даних. Система

управління базами даних (СУБД) витягує інформацію з бази даних у відповідь на запити [21].

База даних зберігається у вигляді файлу або набору файлів на жорсткому диску, чи іншому вторинному пристрої зберігання даних. Інформація в цих файлах може бути розбита на записи, кожен з яких складається з одного або декількох полів. Поля є основними одиницями зберігання даних, і кожне поле, як правило, містить інформацію, що відноситься до одного аспекту або атрибуту сутності, описаної базою даних. Записи також організовані в таблиці, які містять інформацію про взаємозв'язки між різними її областями. Хоча база даних застосовується вільно до будь-якого збору інформації в комп'ютерних файлах, база даних у строгому розумінні забезпечує можливості перехресних посилань. Використовуючи ключові слова та різні команди для сортування, користувачі можуть швидко шукати, переставляти, групувати та вибирати поля у багатьох записах для отримання чи створення звітів про певні сукупності даних.

Записи та файли бази даних повинні бути структуровані, щоб забезпечити можливість отримання інформації. Запити - це головний спосіб, коли користувачі отримують інформацію з бази даних. Потужність СУБД пов'язана з її здатністю визначати нові відносини від основних, заданих таблицями, і використовувати їх для отримання відповідей на запити. Зазвичай користувач надає рядок символів, а комп'ютер здійснює пошук у базі даних відповідної послідовності та надає вихідні матеріали, у яких ці символи відображаються; користувач може запитувати, наприклад, усі записи, у яких вміст поля для прізвища людини – це «Хілько».

Багато користувачів великої бази даних повинні мати можливість в будь-який момент швидко маніпулювати інформацією всередині неї. Більше того, великий бізнес та інші організації, як правило, створюють багато незалежних файлів, що містять пов'язані дані, а їх діяльність з обробки даних часто вимагає зв'язування даних із кількох файлів. Для підтримки цих вимог було

розроблено кілька різних типів СУБД: ієрархічні, мережеві, реляційні та об'єктно-орієнтовані [22].

Записи в ієрархічних базах даних організовані у структурі, подібній до дерева, при цьому кожен рівень записів розгалужується на набір менших категорій. На відміну від ієрархічних баз даних, які забезпечують поодинокі зв'язки між наборами записів на різних рівнях, мережеві бази даних створюють безліч зв'язків між наборами, розміщуючи посилання або покажчики, до одного набору записів в інший; швидкість та універсальність мережевих баз даних призвели до їх широкого використання в бізнесі та в електронній комерції. Реляційні бази даних використовуються, коли асоціації між файлами або записами не можуть бути виражені посиланнями; простий плоский список стає одним рядком таблиці, або "співвідношенням", і кілька відношень можуть бути математично пов'язані, щоб отримати бажану інформацію. Різні ітерації SQL (структурована мова запитів) широко використовуються в СУБД для реляційних баз даних. Об'єктно-орієнтовані бази даних зберігають і маніпулюють складнішими структурами даних, що називаються "об'єктами", які організовуються в ієрархічні класи, які можуть успадковувати властивості від класів вище у ланцюзі; ця структура бази даних є найбільш гнучкою та адаптованою [23].

Інформація у багатьох базах даних складається з текстів документів на природній мові; бази даних, орієнтовані на число, в першу чергу містять інформацію, таку як статистика, таблиці, фінансові дані та необроблені науково-технічні дані. Невеликі бази даних можуть підтримуватися в персональних комп'ютерних системах і можуть використовуватися особами вдома. Ці та більші бази даних набувають все більшого значення у діловому житті, частково через те, що вони зараз створені для інтеграції з іншими офісними програмами, включаючи програми електронних таблиць.

Типові додатки комерційних баз даних включають бронювання авіакомпаній, функції управління виробництвом, медичну документацію в лікарнях та юридичну документацію страхових компаній. Найбільші бази

даних зазвичай підтримуються урядовими установами, бізнес-організаціями та університетами. Ці бази даних можуть містити тексти таких матеріалів, як тези, доповіді, законодавчі статuti, послуги проводів, газети та журнали, енциклопедії та каталоги різного роду. Довідкові бази даних містять бібліографії або покажчики, які служать орієнтирами щодо розташування інформації в книгах, періодичних виданнях та іншій опублікованій літературі. Зараз існують тисячі цих загальнодоступних баз даних, які охоплюють теми, починаючи від права, медицини та інженерії, закінчуючи новинами та поточними подіями, іграми, класифікованою рекламою та навчальними курсами [24].

Все частіше окремі бази даних поєднуються в електронному вигляді в більші колекції, відомі як сховища даних. Підприємства та урядові установи потім використовують програмне забезпечення для «аналізу даних» для аналізу декількох аспектів даних для різних моделей. Наприклад, державне агентство може поставити прапор для людських розслідувань компанію чи фізичну особу, яка придбала підозрілу кількість певного обладнання чи матеріалів, незважаючи на те, що закупівлі поширювалися по країні або через різні дочірні компанії [25].

2.3. Допомога Chrome DevTools у створенні веб-сайту

Chrome DevTools - це набір інструментів веб-розробників, вбудованих безпосередньо в браузер Google Chrome. DevTools може допомогти швидко редагувати сторінки та швидко діагностувати проблеми, що в кінцевому підсумку допомагає швидше створювати веб-сайти [26].

Для того щоб відкрити Інструмент дослідження Google Chrome, потрібно дотримуватися однієї з інструкцій.

З меню Chrome натиснути кнопку меню Chrome у верхньому кутку;

Натиснути «Додаткові інструменти» (More Tools) > «Інструменти розробника» (Developer Tools) (рис.2.1).

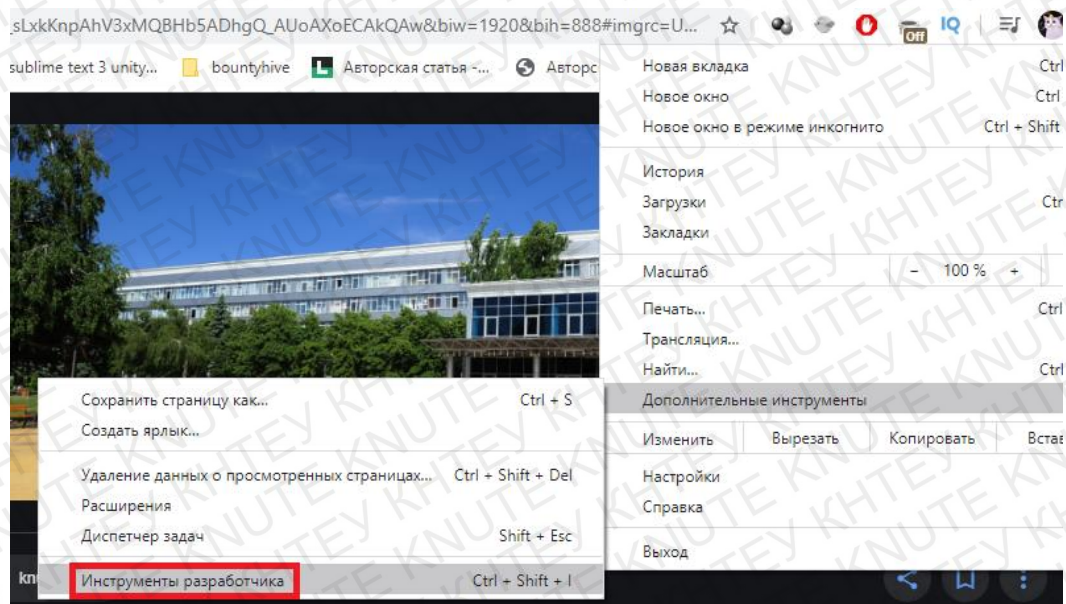


Рис.2.1 Як відкрити інструменти розробника

Натиснути правою кнопкою мишки на елемент, який потрібно дослідити і вибрати «Продивитися код» (Inspect) (рис.2.2)

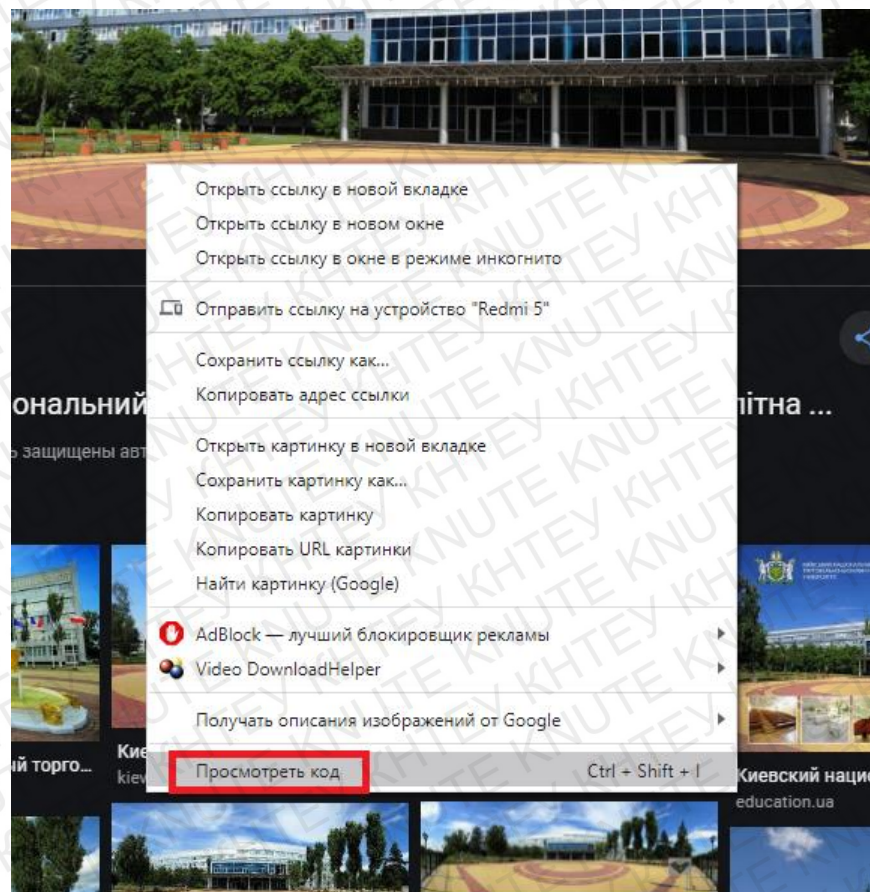


Рис.2.2 Відкриття вікна Dev Tools

Також можна використовувати гарячі клавіші: Ctrl + Shift + I (або Cmd + Opt + I для Mac). В якості альтернативи, можна натиснути на F12 на клавіатурі.

У цьому інструменті є кілька доступних налаштувань:

Редагування стилів CSS. Вкладка Елемент (Element) дозволяє редагувати CSS в живому режимі. Це означає те, що Ви можете попередньо подивитися зміни, які Ви вносите в CSS вашого сайту, перед тим як вносити зміни безпосередньо на сайті.

Тут можна побачити, які стилі закріплені за певними елементами вашого сайту.

Відкривши Інструмент розробника Chrome, використовуючи один з методів, описаних раніше можна побачити стилі CSS сайту.

Тут можна редагувати код CSS, для того щоб дізнатися, як вони будуть відображатися на вашому сайті (рис.2.3).



Рис.2.3 Вікно зі стилями у Dev Tools

Дослідження розмітки HTML. Для того щоб отримати доступ до коду HTML у браузері, потрібно відкрити інструмент розробника Google Chrome в меню браузера, або використовуючи інструмент дослідження елементів.

У лівому широкому розділі, буде код HTML, який використовується на сторінці (рис.2.4).

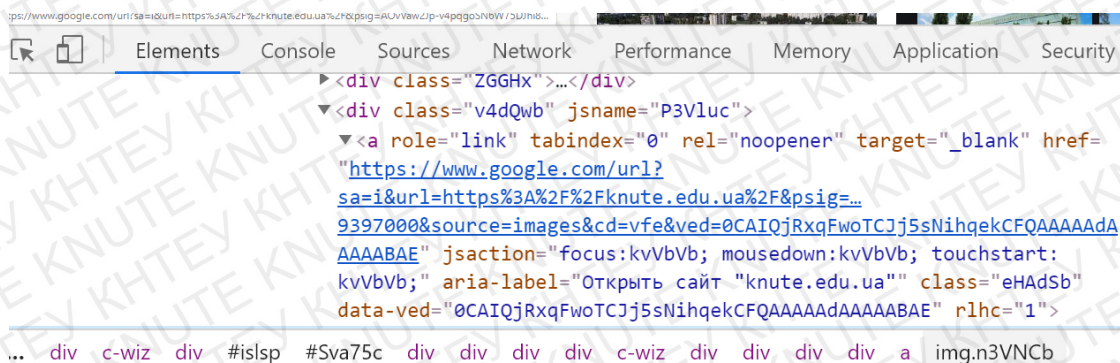


Рис.2.4 Вікно зі кодом html-документу

Вибравши певний елемент сайту, відобразатися буде відповідний код HTML, виділений в браузері. В якості альтернативи, можна натиснути на елемент HTML безпосередньо в Інспекторі, і елемент буде виділений на сторінці.

Робота з Консоллю. Консоль допоможе знайти баги на сторінці. Для того щоб відкрити сторінку Консолі:

Треба використати поєднання клавіш CTRL + Shift + J для Windows (або Cmd + Opt + J для Mac). Або натиснути на кнопку Консоль (Console) в раніше відкритому інструменті розробника (рис2.5).

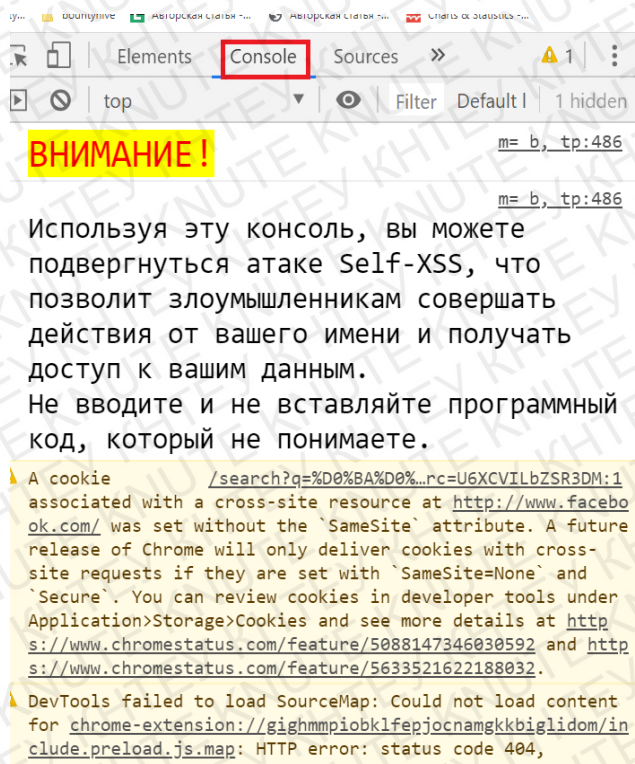


Рис.2.5 Вікно консолі у браузері

Коли Консоль відкрита, будуть відображені помилки, які присутні на сайті.

Цей інструмент дуже корисний, якщо є якісь проблеми з сайтом. Консоль відобразить всі помилки сайту і деякі підказки, як їх усунути.

Також можна використовувати вкладку Мережа, для того щоб побачити файли, які викликають помилки. Помилки відмічені червоним.

Використання функції Емулятора пристроїв Google Chrome

Ця функція допоможе перевірити адаптивність сайту, емулюючи різні розміри екрану і дозволу [27].

Для того щоб отримати доступ до цієї функції, потрібно відкрити інструмент розробника. Натисніть на кнопку (Toggle Device Toolbar) (або використовуйте поєднання клавіш CTRL + Shift + M):

Включивши цей функціонал, можна вибрати мобільний пристрій, на якому потрібно дізнатися, як виглядає сайт при певному розмірі екрану [28].

РОЗДІЛ 3.

СТВОРЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ПІДПРИЄМСТВА ТОРГІВЛІ

3.1 Розробка веб-дизайну (макету) інтернет магазину

При розробці веб-дизайну будь-якого сайту з самого початку потрібно налаштувати усе для роботи.

Потрібно створити файл майбутнього макета і придумати для нього зрозуміле ім'я. Для цього потрібно відкрити Adobe Photoshop і натиснути «Файл - створити» [].

Розміри, які пропонує Adobe Photoshop за замовчуванням, не підійдуть. Потрібно створити файл шириною 1920 пікселів, а потім зберігаємо його, включаємо лінійки «Ctrl + R» і активуємо сітку «Ctrl + E». Якщо все зроблено правильно, то можна буде побачити перед собою лист, розкреслений як шкільний зошит в клітинку (рис. 3.1).

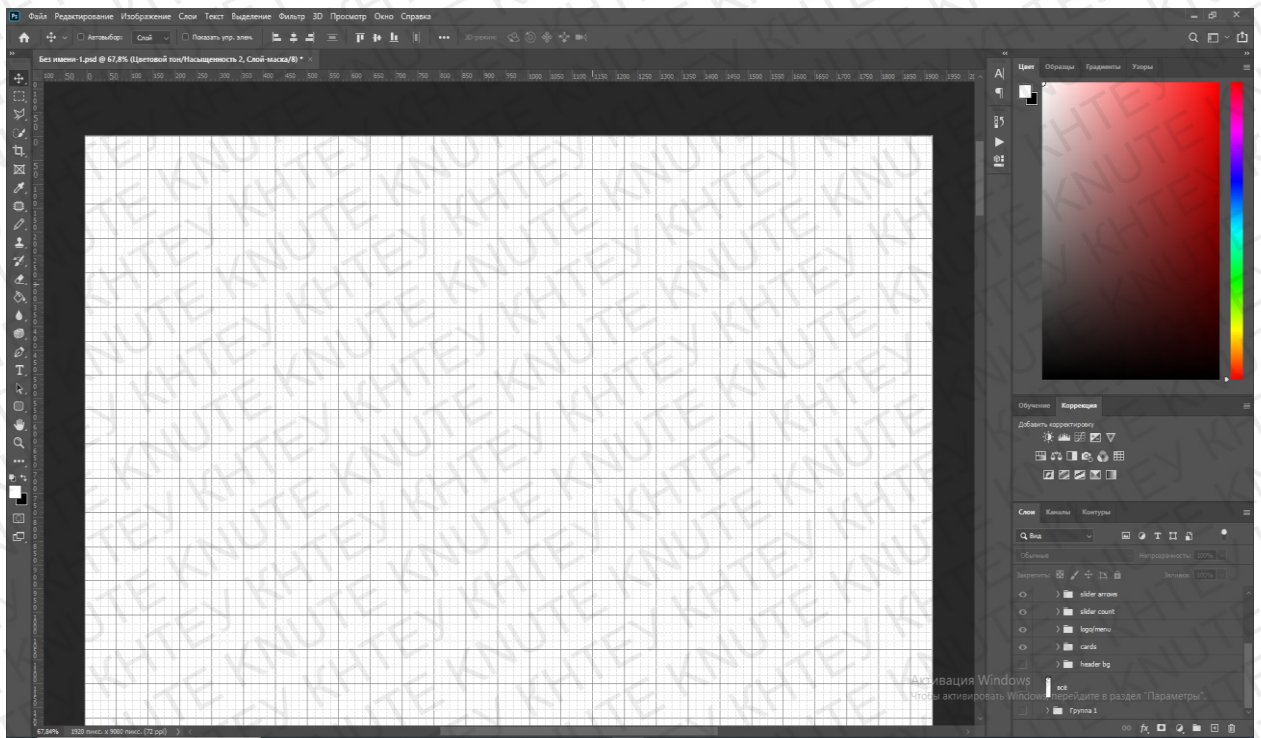


Рис.3.1 Розкреслений лист

На готовому макеті сайту клітинок не буде. Лінії видно тільки під час роботи з макетом і використовуються для того, щоб рівно розташовувати елементи майбутнього шаблону. Сітку можна налаштувати самостійно («Редагування - Налаштування - Спрямовуючі, сітка та фрагменти») або

завантажити готове рішення в Інтернеті. Краще використовувати сітку BOOTSTRAP для зручності подальшої верстки та адаптації

Тепер потрібно встановити спрямовуючі, між якими буде знаходитися Тіло, або основна частина майбутнього ресурсу. Для цього треба натиснути «Перегляд-Нова спрямовуюча» і вибрати орієнтацію «Вертикальна» та вказати положення від лівого краю майбутньої сторінки (по лінійці).

Спрямовуючих потрібно дві, і відстань між ними повинна становити не менше ніж 1003 пікселів(рис.3.2).

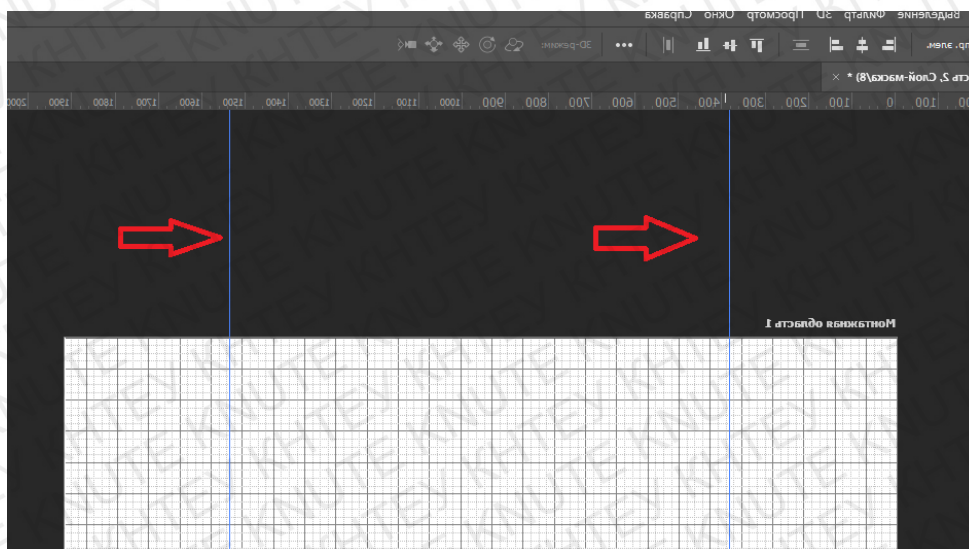


Рис.3.2 Дві спрямовуючі

Після того як робочий простір налаштований можна починати малювати дизайн.

Для початку потрібно задати фон. І не обов'язково малювати картинку самостійно - можна завантажити безкоштовний зображення з будь-якого зі стоків.

Стоки, або фото-стоки - це сайти, на яких можна продавати або розмістити безкоштовно власні зображення та відео, щоб користувачі з усього світу могли використати ці фото [29].

Щоб додати фон, потрібно виділити область бажаного розміру, обрати колір, та зафарбувати виділену область інструментом «Заливка»(рис3.3). Усі потрібні інструменти знаходяться на панелі зліва.

Якщо панель інструментів відсутня, то потрібно ввімкнути її відображення. Для цього потрібно натиснути на пункт «Вид» у «робочій панелі» вгорі, знайти та поставити галочку напроти пункту «панель інструментів».

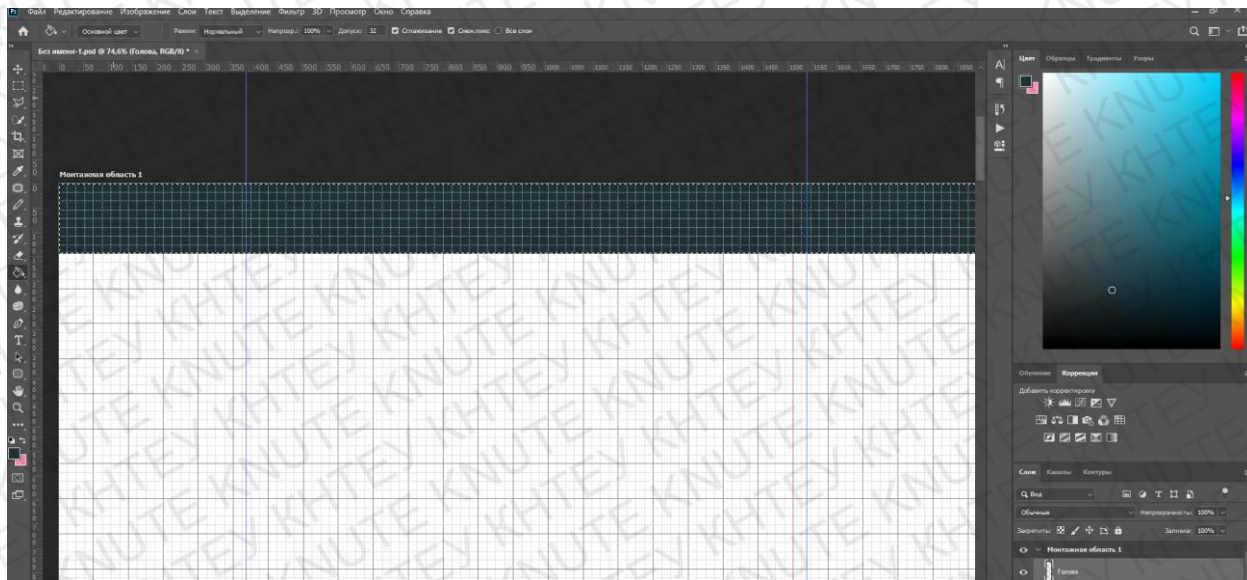


Рис.3.3. Хедер сайту зафарбований обраним кольором

При подальшому створенні дизайну важливо групувати усі шари по групам. Для цього потрібно виділити усі шари певного блоку сайту та натиснути комбінацію клавіш «Ctrl+D»(рис. 3.4).

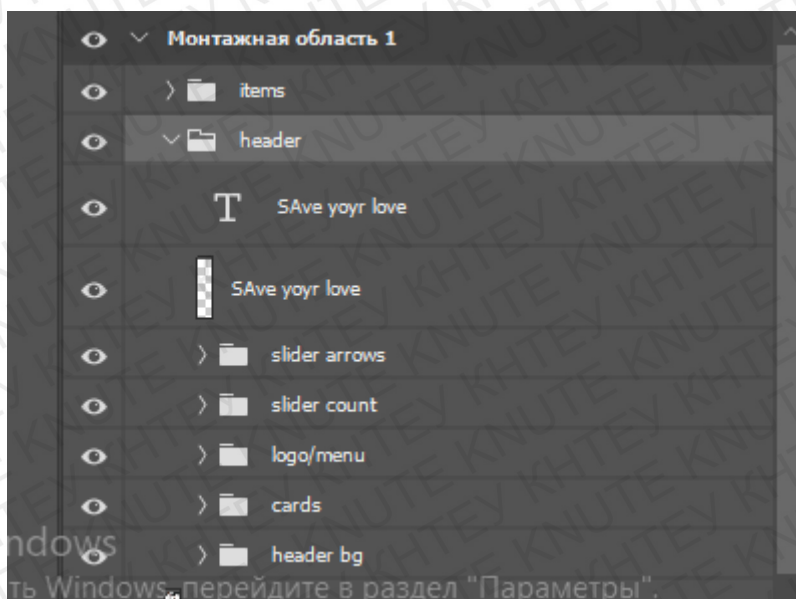


Рис.3.4. Згруповані шари проекту

Для того щоб додати текстові елементи на макет – потрібно обрати інструмент «Текст» та натиснути на макеті туди, де потрібно написати текст. За допомогою цього інструменту був створений логотип.

Далі потрібно знайти картинки майбутніх товарів в інтернеті для того, щоб створити картки товарів. Для більшої привабливості фон потрібно видалити інструментом «Гумка». Після того як фон з усіх фотографій був видалений треба підігнати усі картинки під один розмір за допомогою інструмента «Масштабування».

Усі елементи вирівнюємо опираючись на раніше створені спрямовуючі. Картинки для заднього фону беремо із фото-стоків. Розмістити потрібно використовуючи інструменти описані вище.

Після того як макет буде повністю готовий (рис.3.5) потрібно зберегти проект за допомогою комбінації клавіш «Ctrl + S» та перейти до верстки сайту.

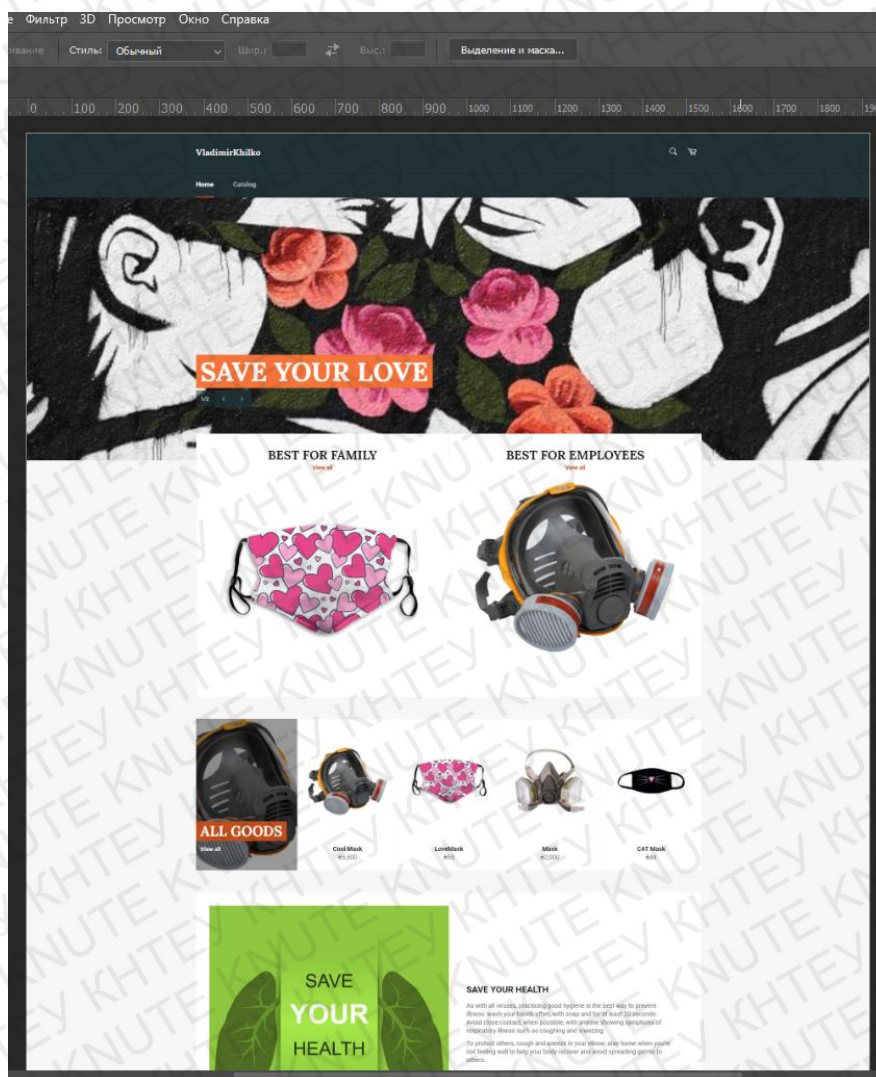


Рис.3.5. Готовий для верстки макет

3.2 Верстка, та налаштування інтернет-магазину

З самого початку потрібно створити файли з назвами: «index.html» та «styles.css» (рис.3.6). У першому буде міститися розмітка написана за допомогою мови гіпертекстової розмітки (HyperText Markup Language — HTML), а в іншому — Cascading Stylesheets (CSS, каскадні таблиці стилів) [30].

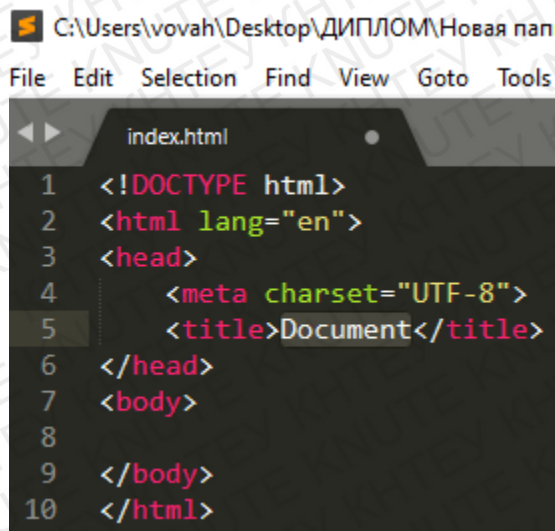


Рис.3.6. Файли для верстки сайту

HTML — стандартна мова розмітки гіпертекстових сторінок в Інтернеті. Є й інші мови розмітки гіпертексту, але значно більша частина сторінок сайтів Інтернету розмічена саме на мові HTML. Такі сторінки успішно інтерпретуються браузерами, які відображають їх на екранах різних електронних пристроїв в зручному для людини вигляді [31].

Якщо бути точним з формальної точки зору, то правильно буде говорити не HTML сторінка, а HTML документ.

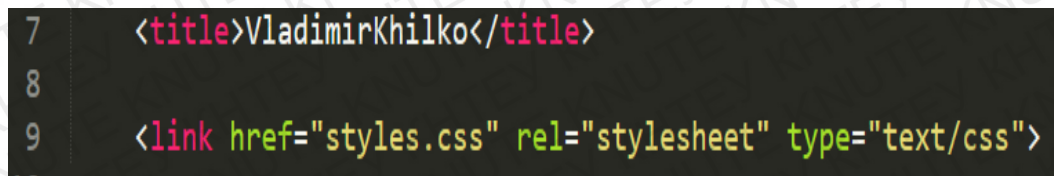
Для кожного HTML документа необхідно вказувати версію, для цього існує спеціальний тег DOCTYPE, який говорить про версії HTML (рис.3.7).



```
C:\Users\vovah\Desktop\ДИПЛОМ\Новая пап
File Edit Selection Find View Goto Tools
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

Рис.3.7. Вказана версія HTML у тегу DOCTYPE

Потрібно прописати заголовок, та підключити файл зі стилями, щоб він працював разом із файлом розмітки. (рис.3.8).



```
7 <title>VladimirKhilko</title>
8
9 <link href="styles.css" rel="stylesheet" type="text/css">
```

Рис.3.8. Код заголовку та підключення файлу стилів

Тег title визначає заголовок сторінки. Це потрібно для того, щоб користувач отримав додаткову інформацію про сайт, щоб краще орієнтуватися при роботі відразу з великою кількістю сайтів.

Тег link призначений для встановлення зв'язку файлу з розміткою з файлами стилів або файлами шрифтів.

Атрибут href визначає напрямок до файлу який підключається. Якщо файли знаходяться в одній директорії, то у атрибут href потрібно записати просто ім'я файлу.

Атрибут rel містить у собі визначення співвідношення файлу та документу у який цей файл підключається. Це потрібно для того, щоб браузер визначив що робити з цим файлом. Значення stylesheet дає зрозуміти браузеру, що підключений файл – це таблиці стилів.

Після того як базові речі були прописані у документі треба будувати структуру сторінки (рис3.9).

Зазвичай це робиться зліва на право, тобто з початку описується шапка веб-сайту починаючи з лівої частини, де частіше за все розташовується логотип, а вже потім описується те що справа. Це не незмінне правило, можна починати писати код з будь-якої частини, але така модель верстки найбільш логічна з точки зору розсудливого програміста. Притримуючись цього правила можна уникнути багатьох плутанин у майбутньому, цим самим скоротити час на верстку.

```
13 <body>
14     <header>
15         <div class="content">
16             <div class="logo">
17                 Vladimikhilko
18             </div>
19             <nav >
20                 <ul >
21                     <li><a href="#"></a></li>
22                     <li><a href="#"></a></li>
23                 </ul>
24             </nav>
25         </div class="content">
26     </header>
27     <div class="content">
28         <cards>
29             <div class="card">
30             </div>
31             <div class="card">
32             </div>
33         </cards>
34     </div class="content">
35 </body>
36 <footer>
37 </footer>
38
```

Рис.3.9. Код структури сторінки

Тег header не має атрибутів. У цьому тегу задають шапку сайту. Тут можна розмістити логотип та меню сайту (nav).

Тег nav прописується для того щоб створити навігацію по сайту. У середині розміщують посилання.

Елемент div являється блоковим, та призначений для того щоб можна було стилізувати вміст всередині конкретного елемента div за допомогою стилів.

Атрибут class являється універсальним та додається до декількох блоків, для того щоб через документ зі стилями можна було змінювати вигляд одразу всіх блоків які потребують однакових значень деяких атрибутів.

Тег footer створюють для того щоб додати туди ім'я автора, інформацію для зв'язку або посилання на соціальні мережі.

Після того як структура була створена потрібно переходити до роботи зі стилями. Прописувати стилі бажано починаючи з верхнього блоку і не переходити до іншого поки всі потрібні стилі не будуть прописані.

```
header {  
    font-size: 26px;  
    top: 0;  
    position: fixed;  
    width: 100%;  
    height: 80px;  
    background-color: #667;  
    justify-content: space-between;  
    border-bottom: 1px solid;  
}
```

Рис.3.10. Код стилів шапки сайту

Атрибут font-size відповідає за розмір шрифту елемента. У цьому випадку встановлене значення 26px. Це означає що текст у тегу header буде рівний 26 пікселям.

Атрибут position зі значенням fixed прикріплює тег header до певної позиції, яку треба вказати атрибутами позиціонування: top, bot left, right. В даному випадку значення атрибуту top прирівнюється до 0. Через це елемент header буде в зафіксованому положенні і при гортанні сторінки завжди буде знаходитися на екрані.

В атрибутах width та height задаються ширина та висота відповідно. Значення ширини 100% означає те, що елемент займе повну ширину батьківського елемента.

Атрибут `justify-content` зі значенням `space-between` розподілить місце порівну між елементами у блоку `header`. Таким чином потрібно побудувати всю сторінку.

При побудові карток з товарами можна додати привабливий для користувача ефект, що спонукає зробити клік по картинці, та дізнатися більше про подібні товари (рисю3.11).

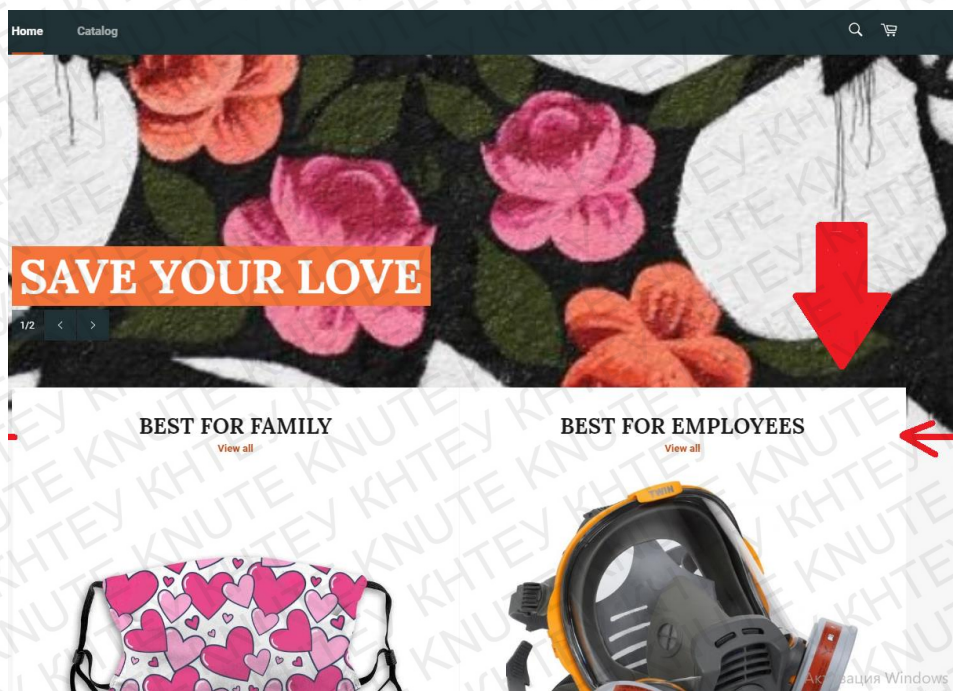


Рис.3.11. Картки товарів з гарним ефектом

Для того щоб досягти цього ефекту потрібно прописати значення `relative` у атрибут `position`. Після цього елементи з цим атрибутом можна буде рухати відносно їх первинного місцезнаходження за допомогою атрибутів позиціонування. Тобто написавши «`top: - 40px`» елемент підніметься на 40 пікселів відносно свого місцезнаходження (рис.3.12).

```
.Cards {  
  position: relative;  
  height: 100%;  
  text-align: center;  
  border-color: #f7f7f7;  
  border-style: solid;  
  border-width: 0 0 1px 1px;  
  background-color: #fff;  
  padding: 0 40px;  
  display: block;  
}
```

Рис.3.12. Код стилю карток з товарами

Атрибут `text-align` зі значенням `center` вирівнює текст по центру батьківського елемента. Як і атрибути: `border-color`, `border-style`, `border-width`, які потрібні для встановлення та стилізації меж, атрибут `text-align` потрібен, щоб вигляд інтернет-магазину був привабливий.

Після того як стилі будуть написані – потрібно завантажити файли на shopify.com для подальшого налаштування магазину (рис.3.13).

[Shopify.com](https://shopify.com) – це сервіс призначений для налаштування функціонування інтернет-магазину. Тут легко адаптувати сайт під мобільні додатки, та під інші пристрої.

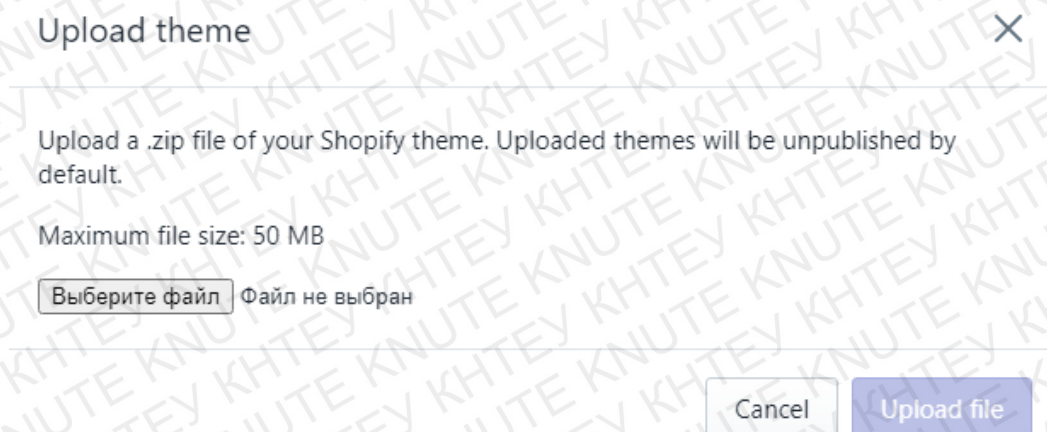


Рис.3.13. Вікно для завантаження свого магазину

Усі скрипти підключаються автоматично (рис.3.14), але щоб вони правильно функціонували нам потрібно вказати які елементи за що

відповідають. Щоб товари можна було додати до кошика, на shopify.com потрібно ці товари створити (рис.3.15).

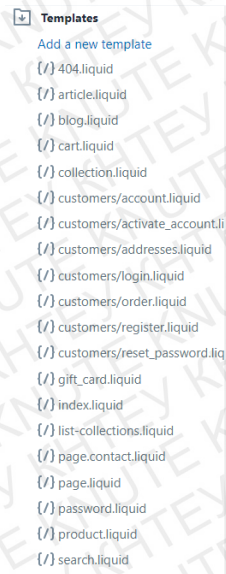


Рис.3.14. Скрипти на сервісі shopify.com

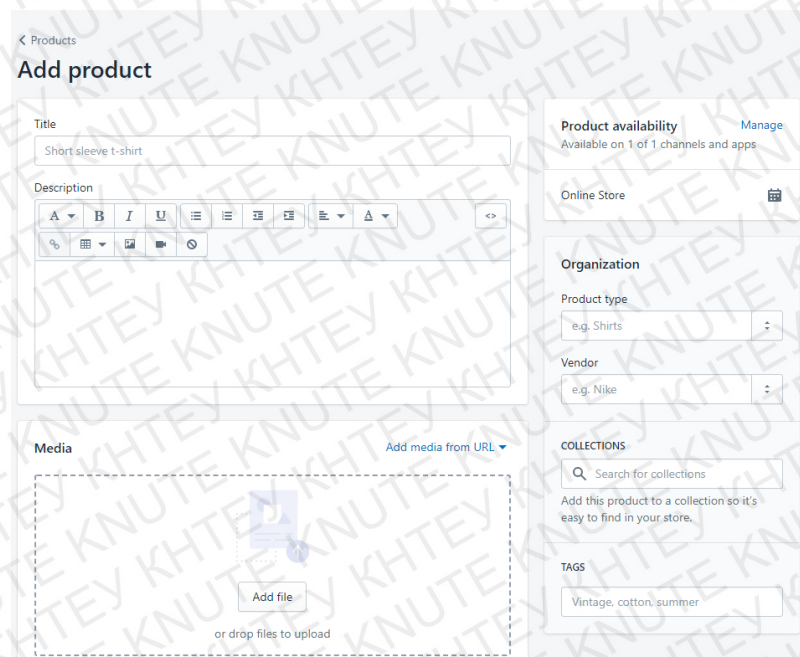
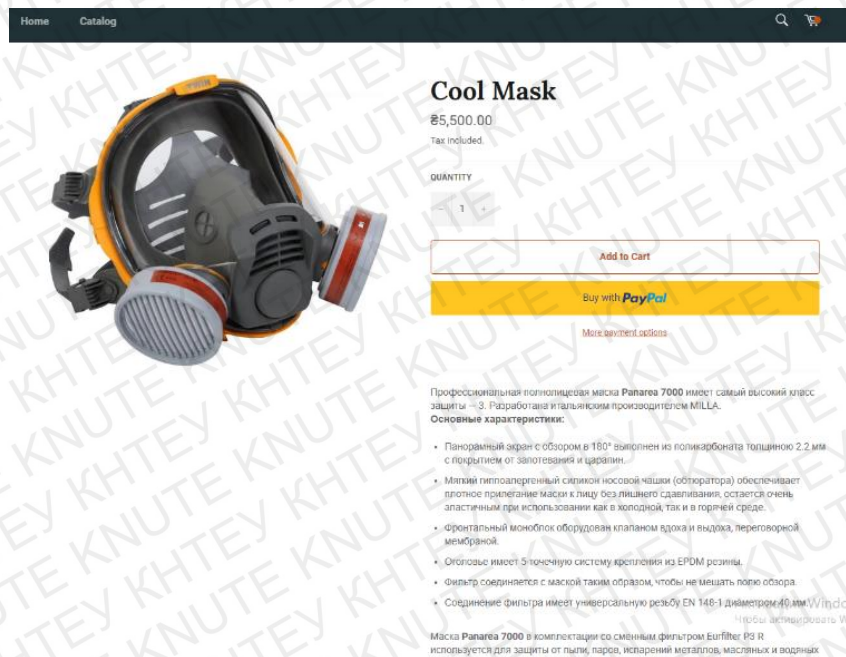


Рис.3.15. Вікно створення товару

Потрібно додати назву, опис та фотографію. Тепер при натисканні на товар, перед користувачем відкриється сторінка з описом товару (рис.3.16).



Home Catalog

Cool Mask

€5,500.00
Tax included.

QUANTITY
1

Add to Cart

Buy with **PayPal**

More payment options

Профессиональная поликарбонатная маска Rapatee 7000 имеет самый высокий класс защиты – 3. Разработана и итальянским производителем MILLA.

Основные характеристики:

- Панорамный экран с обзором в 180° выполнен из поликарбоната толщиной 2,2 мм с покрытием от запотевания и царапин.
- Малый гиппоаллергенный сиддон носовой части (обтуратор) обеспечивает плотное прилегание маски к лицу без лишнего сдавливания, остается очень эластичным при использовании как в холодной, так и в горячей среде.
- Фронтальный моноблок оборудован клапаном вдоха и выдоха, перегородкой мембранной.
- Очистка имеет 5-портовую систему крепления из EPDM резины.
- Фильтр соединяется с маской таким образом, чтобы не мешать полю обзора.
- Соединение фильтра имеет универсальную резьбу EN 148-1 диаметром 40 мм Window

Маска Rapatee 7000 в комплектации со сменным фильтром EuFilter P3 R используется для защиты от пыли, паров, испарений металлов, масляных и водяных

Рис.3.16. Сторінка відкритого товару

Таким чином потрібно додати усі товари, а після цього налаштувати ціну, кількість, вагу та способи доставки.

База даних вбудована в сервіс shopify.com що спрощує роботу з клієнтами.

3.3 Тестування роботи сайту, перевірка усіх функцій

Перед публікацією інтернет-магазину його потрібно перевірити на наявність будь-яких помилок та недоліків. Усі сторінки повинні відкриватися без проблем. Функціонал повинен працювати без збоїв. В іншому випадку це може відштовхнути користувача.

Перш за все потрібно перевірити перехід на усі можливі сторінки. В даному випадку, це усі сторінки товарів, каталог, кошик, головна сторінка, галерея товарів для сім'ї, та галерея товарів для робітників у сфері будівництва.

Тепер треба перевірити усі функції інтернет-магазину.

«Пошук» – дуже важлива функція для сайті. Використання «пошуку» зберігає час користувача. Чим швидше користувач знайде товар, тим вища вірогідність того, що товар буде придбаний.

Наприклад директорів будівельної компанії потрібні маски з панорамним екраном. Щоб знайти таку маску у пошуку потрібно написати «Панорамний екран» (рис.3.17).

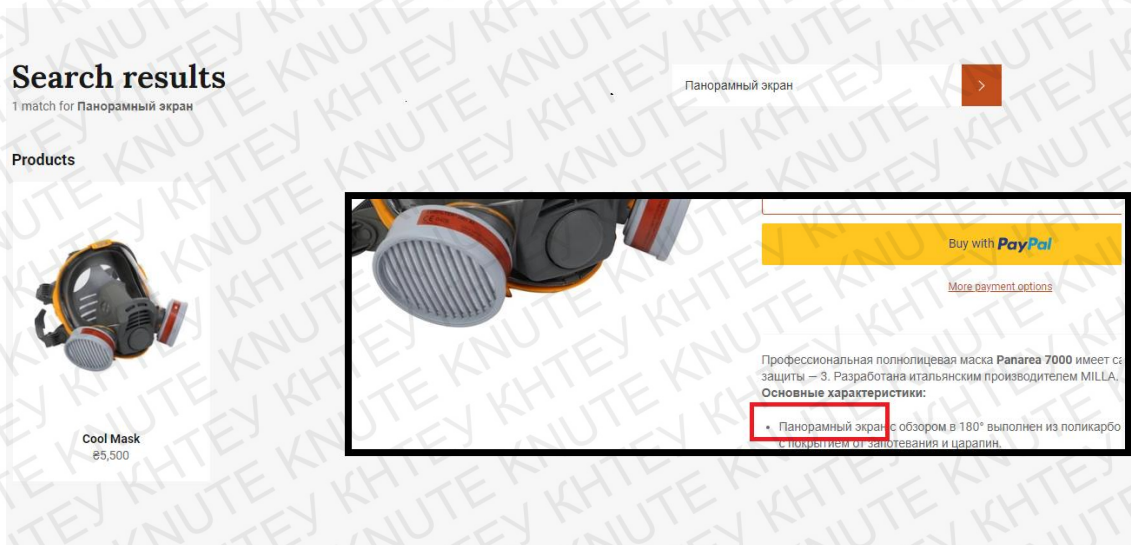


Рис.3.17. Результаты поиска по запросу «Панорамный экран»

Цей товар у своєму описі містить інформацію про наявність панорамного екрану, саме тому «пошук» видає потрібну для користувача інформацію.

Тепер потрібно перевірити чи додаються товари до кошику. Для цього потрібно перейти на сторінку товару, та натиснути на кнопку «Додати до кошику». Після цього потрібно перейти на сторінку кошику та перевірити наявність товару (рис.3.18).



Рис.3.18. Кошик з доданим товаром

Тепер перевіримо сторінку де треба ввести дані для покупки. На цій сторінці потрібно ввести свою контактну інформацію для того щоб продавець міг відправити посилку поштою (рис.3.19).

Contact information

Email or mobile phone number
diplom@mail.ru

Keep me up to date on news and exclusive offers

Shipping address

First name (optional)
Владимир

Last name
Диплом

Address
ул. Киевская 150

Apartment, suite, etc. (optional)

City
Бровары

Country/Region
Ukraine

Postal code
07400

Save this information for next time

[Return to cart](#) [Continue to shipping](#)

Рис.3.19. Сторінка з полем для вводу контактної інформації

Після того як усі дані були введені потрібно перейти на сторінку підтвердження покупки. Тут потрібно ввести дані картки, та натиснути на кнопку підтвердити. Після цього дані перейдуть у базу даних сайту де буде зазначено що покупка оплачена (рис.3.20).

Orders

↑ Export

! You're viewing an updated orders list
Organize work on your orders without needing to leave the page. See more detailed information in one place, including notes, items, and tags.

[Give feedback](#) [Learn more](#)

All Unfulfilled Unpaid Open Closed

Filter orders Status Payment status Fulfillment status Mor

Order	Date	Customer	Order total	Payment	Fulfillment	Items	Delivery method
#1002	Today at 2:58 pm	Владимир Диплом	€55.00	● Paid	○ Unfulfilled	1 item	Standard
#1001	Yesterday at 8:13 pm	ВОВА ХИЛЬКО	€5,500.00	● Paid	○ Unfulfilled	1 item	Standard

Рис.3.20. База даних інтернет-магазину.

У базі зберігаються не тільки вдалі операції. Ведені дані зберігаються після підтвердження введеної контактної інформації.

ВИСНОВКИ

У даному випускному кваліфікаційному проекті представлені результати досліджень технологій та методів створення інтернет-магазину підприємства торгівлі. Був проведений аналіз сучасних інформаційних технологій які стосуються створення сучасного веб-дизайну та верстки сайтів.

Аналіз таких принципів побудови інтернет-магазину як: склад проекту, правильне наповнення сайту, розробка веб-дизайну перед тим, як можна починати верстку, як правильно налаштувати інтернет-магазину, тестування функцій інтернет-магазину надав розуміння загальної картини та масштабів роботи.

У другому розділі були досліджені способи ручної верстки сайту за допомогою HTML та CSS. Аналіз цих способів дав змогу обрати оптимальну стратегію дій, що дозволило, створити сучасний інтернет-магазин, та скоротити час монотонної роботи.

Також були проаналізовані та досліджені функції Chrome Dev Tools, що дозволило швидко знаходити, та виправляти помилки, а також надало можливість зручного редагування коду напряму у браузері.

Аналізи цих методів дозволили обрати ту стратегію, яка була найбільш зручною для створення інтернет-магазину.

Створення інтернет-магазину являється дуже актуальною задачею. Постійний доступ із комп'ютера або мобільного телефону значно підвищує рівень інформованості користувача, що являється плюсом як і для підприємства торгівлі, так і для покупця.

У третьому розділі здійснювалася розробка веб-дизайну інтернет-магазину підприємства торгівлі. На основі проведених аналізів для створення макету була використана програма Adobe Photoshop. Використання даного методу допомогло створити сучасний, приємний та зручний для користувача дизайн.

Завдяки методам чистої HTML та CSS верстки було побудовано сайт, що дозволить будь-якому веб-програмісту легко орієнтуватися в коді та змінювати потрібні елементи.

За допомогою функцій Chrome Dev Tools правильне розміщення елементів на сторінці значно прискорилося, також своєчасно були виправлені помилки.

Результатом усієї кваліфікаційної роботи був створений інтернет-магазин підприємства торгівлі метою якого є збільшення кількості клієнтів, підвищення конкурентоспроможності та підвищення популярності свого бренду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статична сторінка? [Електронний ресурс] – Режим доступу: http://dmytro.github.io/startpack/static_or/
2. Різниця між статичними та динамічними веб-сторінками [Електронний ресурс] – Режим доступу: <https://uk.bccrwp.org/compare/difference-between-static-and-dynamic-web-pages/>
3. Що таке інтернет-магазин [Електронний ресурс] – Режим доступу: <https://i.factor.ua/ukr/journals/nibu/2015/december/issue-101/article-14158.html>
4. Види контенту для сайту. [Електронний ресурс] – Режим доступу: https://ideyne.com/ua/article/vidi_kontenta_dlya_saita_socialnih_setei
5. З чого почати створення сайту? [Електронний ресурс] – Режим доступу: <https://www.taina.com.ua/z-choho-pochaty-stvorennja-sajtu/>
6. Релевантність [Електронний ресурс] – Режим доступу: <https://igroup.com.ua/seo-articles/relevantnist/>
7. Наповнення сайту [Електронний ресурс] – Режим доступу: <https://astwellsoft.com/uk/blog/content.html>
8. Веб-дизайн: переваги і недоліки професії веб-дизайнера. [Електронний ресурс] – Режим доступу: http://mirpiar.com/publ/veb_dizajn_perevagi_i_nedoliki_profesiji_veb_dizajnera/36-1-0-5471
9. Основні етапи створення сайту. [Електронний ресурс] – Режим доступу: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/osnovni-etapy-stvorennia-saitu>
10. Карта сайту – що, як, навіщо. Повне керівництво по Sitemap [Електронний ресурс] – Режим доступу: <https://textum.com.ua/uk/blog/karta-sajtu-shho-yak-navishho-povne-kerivnitstvo-po-sitemap/>
11. Етапи створення веб-сайтів [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/siteingschool/etapi-stvorennia-veb-sajtiv>

12. Розробка дизайну сайту [Електронний ресурс] – Режим доступу: <https://studio-creative.com.ua/uk/dyzajn/dizajn-sajta/>
13. Основи HTML [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/HTML_basics
14. Мова JavaScript та її можливості [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/webtehnologiiaetawebdizajn/mova-javascript-ta-ieie-mozlivosti>
15. Веб-програмування. Серверні скрипти [Електронний ресурс] – Режим доступу: <https://webstudio2u.net/ua/programming/144-server-scripts.html>
16. Що таке верстка сайту? [Електронний ресурс] – Режим доступу: <https://venweb.com.ua/shcho-take-verstka-sajtu/>
17. Лекція 3 - CSS, HTML Таблична верстка [Електронний ресурс] – Режим доступу: <http://fcit.tneu.edu.ua/web-rozrobka/html-css/lektsiia-3-css-html-tablychna-verstka>
18. 4 ОСНОВНИХ НЕДОЛІКА ТАБЛИЧНОЇ ВЕРСТКИ [Електронний ресурс] – Режим доступу: <http://web.if.ua/65-4-osnovnih-nedolka-tablichnoyi-verstki.html>
19. Блокова верстка [Електронний ресурс] – Режим доступу: <http://lesson-css.blogspot.com/p/bloкова-verstka.html>
20. Блокова верстка сайту - верстка div [Електронний ресурс] – Режим доступу: <https://webstudio2u.net/ua/programming/428-div-deveorment.html>
21. Що таке база даних? [Електронний ресурс] – Режим доступу: <http://areps.kpi.ua/shco-take-basa-danykh>
22. Основні відомості про бази даних [Електронний ресурс] – Режим доступу: <https://studfile.net/preview/5784260/>
23. 11 типів сучасних баз даних: короткий опис, схеми і приклади БД [Електронний ресурс] – Режим доступу: <https://senior.ua/articles/11-tipv-suchasnih-baz-danih-korotkiy-opis-shemi--prikлади-bd>

24. бази даних та інформаційно-довідкові системи [Електронний ресурс] – Режим доступу: <https://ukrpatent.org/uk/articles/bases2>
25. Класифікація баз даних [Електронний ресурс] – Режим доступу: <http://lib.mdpu.org.ua/e-book/vstup/L5.htm>
26. Chrome DevTools: налаштування, можливості та способи перевірки коду [Електронний ресурс] – Режим доступу: <https://dou.ua/lenta/articles/chrome-dev-tools-guide/>
27. Огляд дванадцяти можливостей Chrome Devtools [Електронний ресурс] – Режим доступу: <https://codeguida.com/post/573>
28. Практичний огляд інструментів розробника у найпопулярніших браузерах [Електронний ресурс] – Режим доступу: <https://www.quality-assurance-group.com/oglyad-instrumentiv-rozrobnyka-u-najpopulyarnishyh-brauzerah/>
29. Фотостоки. Що це таке і з чим це їдять. [Електронний ресурс] – Режим доступу: <http://martatrotsiuk.com/archives/1153>
30. Основи CSS. [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/CSS_basics
31. HTML. [Електронний ресурс] – Режим доступу: <https://developer.mozilla.org/uk/docs/Web/HTML>