

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ РЕКОМЕНДАЦІЙ
СИСТЕМИ ДЛЯ ОНЛАЙН МАГАЗИНУ**

Здобувач освіти гр. ІН.м-12ан/2у

Анастасія ФУРСА

Науковий керівник,
к.т.н., доцент

Наталія БАРЧЕНКО

В.о.завідувача кафедри
кандидат технічних наук, доцент,
доцент кафедри комп'ютерних наук

Ігор ШЕЛЕХОВ

Суми 2022

Сумський державний університет

(назва вузу)

Факультет ЕЛПІ Кафедра Комп'ютерних наук

Спеціальність «122 – Комп'ютерні науки»

Затверджую:

зав.кафедрою _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Фурса Анастасія Андріївна

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія проєктування рекомендацій системи для онлайн магазину

затверджую наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи)

3. Вхідні данні до проекту (роботи)

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми. Огляд існуючих рішень. Огляд алгоритмів 2) Постановка задачі й формування завдань дослідження. 3) Вибір методів рішення задачі. 4) Моделювання та проєктування системи. 5) Практична реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	<i>Аналіз проблеми. Огляд існуючих аналогів. Огляд алгоритмів</i>	<i>01.09 – 12.09</i>	
2.	<i>Постановка завдання та вибір методів реалізації</i>	<i>15.09 – 26.09</i>	
3.	<i>Моделювання та проєктування веб-додатку.</i>	<i>29.09 – 30.10</i>	
4.	<i>Практична реалізація.</i>	<i>03.11 – 28.11</i>	
5.	<i>Оформлення пояснювальної записки до дипломної роботи</i>	<i>01.12 – 10.12</i>	

Студент – дипломник

(підпис)

Керівник проекту

(підпис)

РЕФЕРАТ

Записка: 62 стор., 38 рис., 6 додатків, 15 літературних джерел.

Об'єкт дослідження – інтрнет-магазин з алгоритмом пошуку та відображення рекомендованих товарів.

Мета роботи – розробка інтернет-магазину з електроніки та підбору рекомендованих товарів з використанням сучасних веб-технологій.

Результати – реалізовано інтернет-магазин електроніки з алгоритмом підбору рекомендованих товарів з використанням HTML+CSS+JS та мови програмування PHP та СУБД MySQL.

ВЕБ-СЕРВІС, ВЕБ-РОЗРОБКА, ІНТЕРНЕТ-МАГАЗИН, ЕЛЕТКРОНІКИ, АЛГОРИТМ, ІНФОРМАЦІЙНА СИСТЕМА, АНАЛІЗ.

ЗМІСТ

РЕФЕРАТ.....	5
ВСТУП.....	7
1. ІНФОРМАЦІЙНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	8
1.1 Огляд проблемної області.....	8
1.2 Огляд подібних рішень	8
1.3 Створення сайтів	9
1.4 Огляд подібних рішень	11
1.5 Постановка задачі.....	15
2. ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ	19
2.1 Вибір інструментальних засобів розробки.....	19
2.2 Бібліотеки JavaScript	19
2.3 Бібліотеки CSS.....	23
2.4 Інтеграція стека технологій	25
2.5 Опис середовища та системи.....	27
3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ЕЛЕКТРОНІКИ.....	32
3.1 Розробка моделі інформаційної системи	32
3.2 Програмна реалізація	36
3.3 Тестування	42
ВИСНОВКИ	48
СПИСОК ЛІТЕРАТУРИ.....	49
ДОДАТОК	50
Додаток А(header.php).....	50
Додаток Б(footer.php)	51
Додаток В(registration.php).....	52
Додаток Г (index.php).....	53
Додаток Е (script.js)	55
Додаток Ж(style.css)	60

ВСТУП

У сучасному світі людині майже не вистачає часу, бажання, а іноді і фізичних сил, аби відвідати необхідний магазин. Найпростіший варіант – відкрити веб-сервіс і вибрати потрібний товар. Сучасні інтернет-магазини, наприклад Lamoda, дозволяють вибрати певну кількість розмірів для доставки, щоб клієнт мав право приміряти дрібницю і обрати ту, що пасує найбільше. На сьогоднішній день існує велика кількість інтернет-магазинів різного роду спеціалізацій, наприклад: Lamoda, Aliexpress, Amazon, Ebay. Вибір веб-сервісів також різноманітний, наприклад: господарські магазини, магазини одягу, спортивні магазини, вузькоспеціалізовані магазини і т.д. Для залучення клієнтів інтернет-магазини організовують велику кількість акцій і проводять величезні рекламні кампанії. У разі успішної рекламної кампанії їх прибуток зростає.

Зараз розробка веб-сайтів є досить поширеною діяльністю саме тому велика кількість людей працює у цій сфері. Найбільшим досягненням у фактичній технології розробки веб-сайтів є розробка веб-сервісів. Інструменти управління контентом широко використовуються в Інтернет-просторі при розробці веб-сайтів будь-якої складності.

Основною відмінною рисою систем управління контентом, яка водночас є і перевагою, є можливість розробляти на практиці сайти будь-якого рівня складності з використанням HTML, CSS та інших областей веб-програмування. А також можливість практично миттєво, легко та інтуїтивно створювати, видаляти, редагувати та формувати контент, це значно спрощує завдання адміністрування сайту. Таким чином, завдання розробки інформаційної системи для магазину електроніки з реалізацією алгоритму підбору рекомендованих товарів є актуальним.

Реалізація поставленої задачі автоматизує наявні процеси та забезпечить: структуроване збереження даних, їх опрацювання та пошук. Використання такого програмного продукту надасть можливість співпрацювати з адміністрацією інтернет-магазину у зручний для користувача спосіб.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд проблемної області

Метою представленого випускного проекту є створення інтернет-магазину з продажу електроніки в мережі Інтернет.

Завданнями випускного проекту, відповідно до мети, є:

1. Аналіз та опис предметної області;
2. Постановка завдання на створення інтернет-магазину;
3. Створення методу вирішення проблеми створення веб-сервісу;
4. Підбір програмних засобів для реалізації створення інтернет-магазину;
5. Презентація та опис архітектури веб-сервісу;
6. Програмна реалізація інтернет-магазину.

Практичне значення даного випускного проекту – розробка інтернет-магазину, який після встановлення та розміщення, а також наповнення товарами та послугами почне повноцінно виконувати свої цілі, функції та завдання.

Ці функції включають:

1. Можливість створювати, переглядати та редагувати категорії товарів і послуг;
2. Можливість редагування, обслуговування замовлення;
3. Право вибору видів і способів оплати товарів і послуг;
4. Право вибору видів і способів доставки;
5. Наявність панелі керування адміністратора веб-сервісу.

Основними способами виконання завдання є знаходження та аналіз подібних проектів за темою, визначення особливостей та відмінних рис власного інтернет-магазину, його програмна реалізація, впровадження та підтримка.

1.2 Огляд подібних рішень

Інтернет-магазин – це веб-сайт, який рекламує товари чи послуги, приймає та обробляє замовлення на покупку певних товарів чи послуг, надає

користувачеві можливість вибору варіанту оплати, виду покупки та оформлення замовлення.

Запровадження веб-сервісу має такі цілі:

1. Збільшення ринку та збільшення кількості клієнтів;
2. Зниження витрат на оплату праці менеджерів з продажу;
3. Зниження витрат на постачання основних видів діяльності (збут);
4. Зменшення кількості помилок у звітності та збільшення швидкості її

складання.

Створений інтернет-магазин зобов'язаний забезпечити виконання абсолютно всіх звичних для таких проектів функцій, а саме:

1. Можливість реєстрації користувачів;
2. Можливість створення замовлення з будь-якої кількості товарів і послуг;
3. Наявність багаторівневих категорій товарів і послуг з простим і зручним пошуком за вказаними критеріями;
4. Особистий кабінет зареєстрованих користувачів;
5. Кошик, куди будуть додані товари та послуги, обрані клієнтами;
6. Реєстрація та управління замовленнями, товарами та послугами адміністрацією веб-сервісу;
7. Особистий кабінет користувача з відображенням замовлень;
8. Можливість писати коментарі та відгуки про інтернет-магазин та існуючі товари.

1.3 Створення сайтів

З моменту створення всесвітньої мережі пройшло багато років. З часом багато що змінилося: зріс технічний потенціал Мережі, збільшилась інтернет-аудиторія, завдання та цілі, заради яких створювалася Всесвітня павутина. Інтернет перестав існувати лише в наукових і військових цілях. Сьогодні Інтернет – це великий склад, де кожен може знайти потрібний текстовий документ, музику чи виконавця, відео чи наукову літературу, а також придбати

усе необхідне – від продуктів харчування до програмного забезпечення та комп'ютерних компонентів. Компанії твердо і чітко вважають за необхідне розробити мінімальне інтернет-представництво організації, іншими словами, все більше і більше комерційних сайтів, основною метою яких є розвиток і продаж товарів і послуг через Інтернет [1].

Хоча вже можна сказати, що через деякий час конкурентоспроможність у роздрібній торгівлі через Інтернет буде такою ж інтенсивною, як і у звичайному роздрібному.

Однак що таке інтернет-магазини, як вони розвиваються, які в них недоліки і переваги?

Спочатку інтернет-магазин – це сайт, який має великий каталог з категоріями товарів, описами та зображеннями до них. Основна відмінність від простого онлайн-каталогу в тому, що представлені в веб-сервісі товари можна не тільки подивитися, але і зробити замовлення у будь-якомі місці та у зручний для користувача час.

Існують різні види веб-сервісів:

- Сервіс, що реалізує певний вид товарів чи послуг;
- Сервіс, що реалізує продукцію вузькоспрямованих категорій;
- Електронні гіпермаркети;
- Великі роздрібні мережі, що складаються з веб-сервісів [2].

Для всіх веб-сервісів характерний певний базовий набір компонентів, таких як:

- Загальний каталог з категоріями, які показують усі наявні товари. Зовнішній вигляд каталогів може бути різним – деревоподібні, спливаючі або вкладені списки категорій;

- Процедура реєстрації користувача, яка створить для кожного нового клієнта його особистий кошик, куди можна додати обраний товар чи послугу, а потім замовити. У міру переміщення покупця по каталогу система також буде відслідковувати побажання покупця, на основі чого згодом можна буде будувати

не тільки колекції магазину, а й архітектуру каталогу супутніх товарів. Наприклад, якщо людина вказала книгу у своєму замовленні, система може непомітно рекомендувати переглянути інші книги з тієї ж серії чи автора, або запропонувати прочитати рецензії тих, хто її вже читав. Система може повідомити, що особа, яка замовила подібний товар, також зацікавилася супутніми товарами або послугами - наприклад, при покупці мобільного телефону вам пропонується підібрати до нього чохол в залежності від обраної моделі;

– Система оплати товарів і послуг: клієнту пропонується використовувати різні способи оплати товарів і послуг - безготівковий розрахунок, кредитні картки, оплата електронними грошима, оплата готівкою після доставки (кур'єром або при отриманні товару);

– Система доставки товарів і послуг також характеризується широким спектром опцій: відправка EMAIL (програмне забезпечення, ліцензійний ключ продукту), доставка кур'єрськими чи службою або звичайною поштою.

Однак, не зважаючи уваги на загальні особливості, кожен інтернет-магазин є унікальним і відрізняються один від одного. Власник будь-якого інтернет-магазину прагне зробити свій сайт максимально комфортним для користувача, вдосконалюючи способи переходу з одного каталогу в інший, а також створюючи простий і зрозумілий інтерфейс. Як і в звичайному магазині, в інтернет-ресурсі є можливість влаштовувати знижки та розпродажі. Основна відмінність звичайного магазину від інтернет-магазину – це можливість витратити менше грошей, а також можливість зробити покупку, не відволікаючись від важливих справ і головне, не виходячи з дому чи роботи. Завдяки цьому покупка в Інтернеті стає кращою та більш зручною.

1.4 Огляд подібних рішень

Для розробки інтернет-магазину необхідним є визначення архітектури створюваного веб-сервісу, розуміння принципу його роботи, оформлення навігаційного меню тощо. Важливим є дізнатися інформацію про каталоги

товарів і послуг, розташування, експозицію, а також новини, пов'язані з продукцією компанії, що присвячені певній тематиці, наявність певних онлайн-сервісів (наприклад, онлайн-оплата, замовлення доставки товарів чи послуг), посилення на контактну інформацію, інформацію про компанію та методи її роботи. Крім усього перерахованого вище, в веб-сервісах відображаються статті та новини, пов'язані з проданими товарами чи наданими послугами. Давайте детальніше розглянемо кілька великих веб-сервісів.

В якості прикладів для розробки інтернет-магазину були розглянуті наступні сайти:

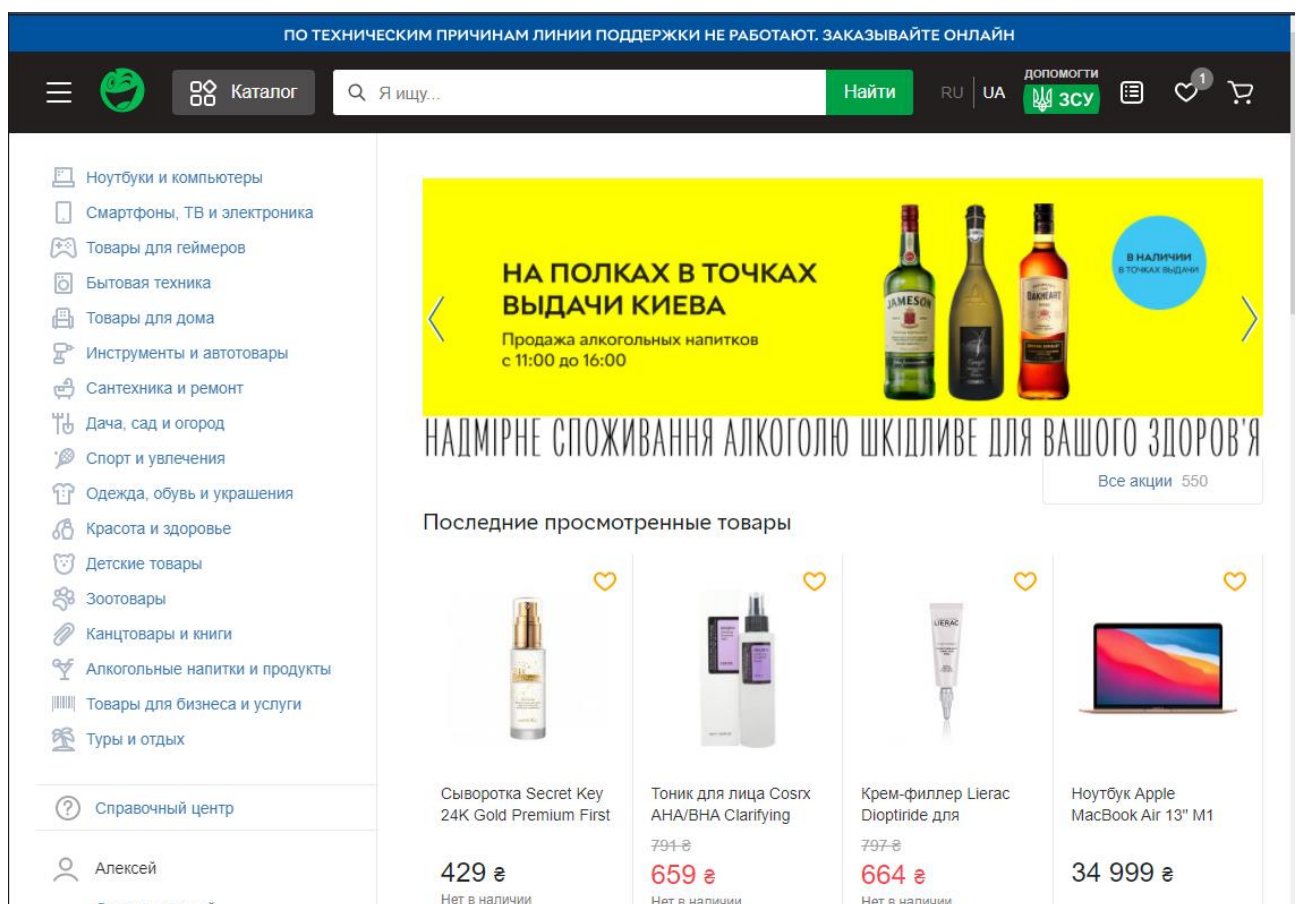


Рисунок 1.1 – Интернет-магазин «Розетка»

Розетка – український веб-сервіс, основний напрямок якого полягає в продажі техніки.

Нижче наведено приклади зарубіжних веб-сервісів.

Наступний веб-сайт, що було обрано для аналізу – це Aliexpress, не менш великий і масштабний інтернет-магазин. Aliexpress вважається одним з найбільших інтернет-магазинів Китаю. Основна мета цього веб-сервісу - продаж товарів і послуг. Коротко розглянемо основні елементи та архітектуру веб-сервісу.

- На основній сторінці (Рис. 1.2) відображаються головне меню, що включає в себе популярні категорії та підкатегорії, а також нові, рекомендовані та популярні продукти чи послуги;

- У нижньому колонтитулі сторінки є посилання для встановлення зворотнього зв'язку, важливі розділи щодо купівлі товарів, контактна інформація та інші найбільш популярні статті;

- Головне меню для пошуку та вибору товарів – це навігаційне меню по категоріям товарів, натиснувши на яке клієнт може легко знайти потрібну категорію товару чи сам товар.

- Також на головній сторінці є форма входу для зареєстрованих клієнтів, після авторизації яким буде доступний особистий кабінет та кошик із вибраними товарами, якщо такий є. Розробником цього інтернет-магазину є Джек Ма, який є засновником Alibaba групи в 1999 році. Сьогодні компанія має власну систему електронних платежів Alipay, dt, Taobao.com та AliExpress.

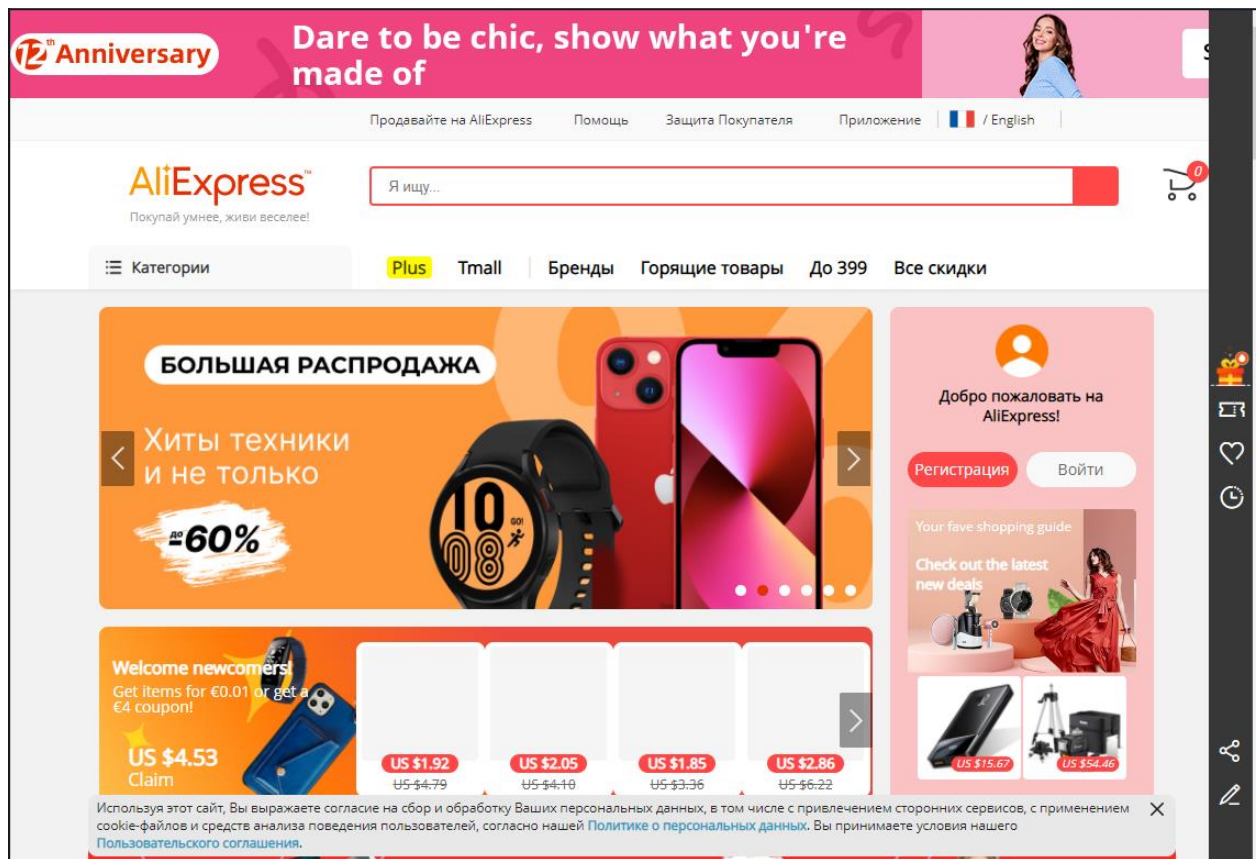


Рисунок 1.2 – Интернет-магазин AliExpress

eBay – цей веб-сервіс присвячений продажу товарів і послуг в Інтернеті. Саме на цій платформі користувачі з усього світу можуть розміщувати власні товари. Найбільше число продавців знаходиться в США, але eBay активно розвивається і в інших країнах. Є такі розділи:

- Основний містить нові та рекомендовані товари, перехід за категоріями товарів, наприклад: електроніка, мода, краса та здоров'я. Також є посилання на інші ресурси (включаючи онлайн-сервіси);
- Крім цього, сайт містить інформацію про платформу, контактні дані, способи оплати та всю необхідну додаткову інформацію, необхідну для покупця;
- У кожному каталозі розташовані підкатегорії для зручності користувачів у навігації та пошуку потрібних товарів;
- Реєстраційні посилання, карта сайту тощо розташовані в нижній частині сторінки. Цей веб-сервіс був розроблений 4 вересня 1995 року в Сан-

Хосе (Каліфорнія) програмістом П'єром Омідьяром, який створив онлайн-аукціон під назвою AuctionWeb як частину свого особистого веб-сайту.

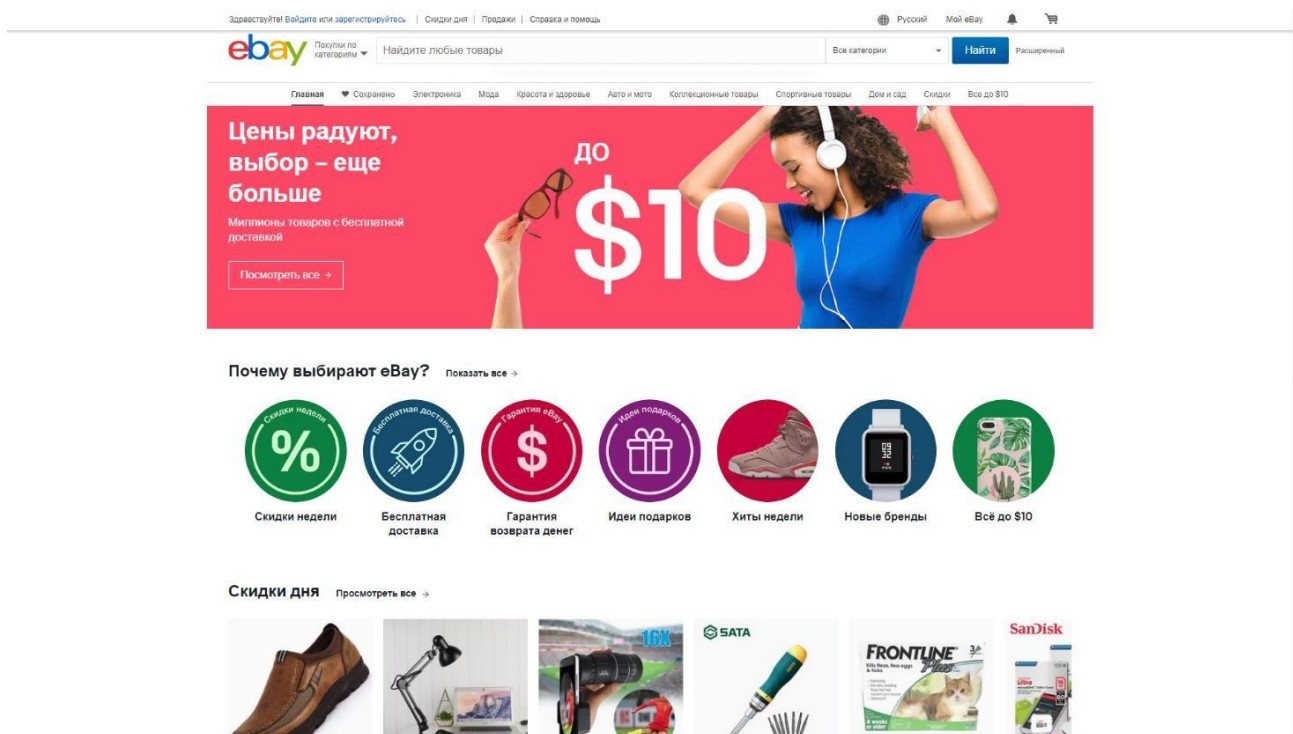


Рисунок 1.3 – Интернет-магазин eBay

1.5 Постановка задачі

Метою дипломного проекту є розробка зручного та простого веб-сервісу з продажу електроніки для збільшення прибутку компанії, розширення клієнтської бази та вихід бізнесу на новий рівень.

Основна мета програми – частково автоматизувати процес пошуку, покупки та замовлення товарів. Така система, з високим рівнем впровадження, повинна полегшити роботу працівників магазину.

Система повинна передбачати поділ прав доступу і бути реалізована для трьох категорій користувачів: співробітники магазину, клієнти, відвідувачі сайту. Слід зазначити, що відвідувачі сайту будуть незареєстрованими користувачами в системі і мати мінімум привілеїв.

Система для відвідувачів (гостей) сайту повинна надавати можливість:

- Зареєструватися в системі;

- Здійснювати пошук товарів за вибраними фільтрами;
- Переглянути інформацію про товар;
- Переглянути список коментарів до товару;
- Зручний, інтуїтивно зрозумілий інтерфейс з підказками;
- Додати товари до кошику.

Частина системи, призначена для покупця, у свою чергу повинна забезпечувати наступні можливості:

- Аутентифікація користувача в системі;
- Перегляд списку товарів у кошику та видалення товарів із кошика, зміна кількості товарів у кошику;
- Оформлення замовлення на покупку товарів в кошику;
- Переглядати та змінювати свій профіль;
- Перегляд історії замовлень, зроблених користувачем;
- Скасування замовлення, зробленого покупцем;
- Зміна пароля користувачем;
- Додавання коментарів;
- Всі можливості, надані відвідувачам (гостям) сайту.

Система для працівників магазину повинна забезпечувати наступні можливості:

- Блокування доступу до системи для інших користувачів;
- Можливість додавати, видаляти, змінювати інформацію про категорії;
- Можливість додавати, видаляти, змінювати інформацію про товари;
- Отримання текстової та графічної інформації про історію покупок;
- Всі можливості, надані покупцям товарів.

При введенні та редагуванні даних система повинна контролювати правильність введеної інформації та, по можливості, виключати ситуації, які можуть призвести до помилкових дій з боку користувачів.

З метою полегшення підтримки та обслуговування програми, а також її подальшого розширення, вона повинна бути розроблена та створена на основі трирівневої архітектури (клієнт, сервер додатків і база даних) для побудови програмних систем, а також мати клієнтську і серверну частини. Серверна частина повинна відповідати за доступ до даних і містити бізнес-логіку програми. Клієнтська частина, у свою чергу, повинна реалізувати користувацький інтерфейс програми.

Необхідним є використання MySQL як рівень доступу до даних, який забезпечує централізоване структуроване зберігання всіх системних даних, гарантуючи їх цілісність і узгодженість, а також надає безліч низькорівневих сервісів для: читання даних зі сховища, збереження даних, зміни їх структури і т.д. На цьому рівні має бути створена база даних, яка буде зберігати всі дані системи. Реалізація команд отримання даних, контроль цілісності та узгодженості даних повинні здійснюватися за допомогою відповідних збережених процедур, тригерів та інших об'єктів, наданих сервером.

Рівень бізнес-логіки буде розгорнутий на сервері додатків і представлятиме ядро системи. Більша частина бізнес-логіки системи повинна бути зосереджена на цьому рівні:

- Алгоритми авторизації користувачів системи, перевірки прав доступу;
- Правила обробки даних, такі як: перевірка правильності заповнення даних користувачем, перевірка та організація взаємозв'язків даних;
- Клас для підключення до бази даних і виконання транзакцій;
- Класи та алгоритми роботи з таблицями БД та запуску виконання відповідних збережених процедур і функцій на сервері.

На рівень презентації необхідно перенести найпростішу бізнес-логіку: інтерфейс авторизації, інтерфейс для відображення товарів, інтерфейс для відображення кошика користувача, інтерфейс для відображення профілю користувача, різноманітні операції з товарами.

Частина інтерфейсу, розроблена в рамках цього проекту, має бути реалізована з використанням мови гіпертекстової розмітки HTML версії 5, а також технології каскадної таблиці стилів CSS та мови сценаріїв браузера JavaScript.

Система повинна бути розроблена на платформі в інтегрованому середовищі розробки Sublime Text.

2. ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Вибір інструментальних засобів розробки

Технологічний стек, що використовується у процесі розробки даного інтернет-магазину, включає в себе фреймворки та бібліотеки, необхідні для ефективною та швидкою розробки. Дані фреймворки та бібліотеки дозволяють з мінімальною кількістю даних та обмеженими системними вимогами інтегрувати веб-додаток з іншими службами для відображення даних у WEB інтерфейс в режимі реального часу.

2.2 Бібліотеки JavaScript

Розглянемо основні бібліотеки та фреймворки JavaScript, які використовуються в сучасному веб-програмуванні. У роботі [5] порівнюється їх продуктивність (згідно Рис. 2.1) у браузері Google Chrome 48. У даному випадку було створено 1000 рядків відразу після завантаження сторінки («створити 1000 рядків»), оновлюючи 1000 рядків у таблиці після 5 ітерацій «розігрів» движка JavaScript («оновити 1000 рядків (гаряче)») часткове оновлення рядків у таблиці після 5 ітерацій «розігріву» движка JavaScript (додавання крапки в кінець кожного 10-го рядка, «часткове оновлення»), виділення рядка після 5 ітерацій «прогрівання» движка JavaScript (візуальне виділення рядка, «вибрати рядок»), видалення рядка після 5 ітерацій «розігріву» движка JavaScript («видалення рядка»). Відповідно до гістограми (Рис. 2.1), бібліотека VueJS має найкращі результати, за винятком оновлення рядків у таблиці. Веб-інтерфейс «тестування у психолога» передбачає візуальне оновлення даних в режимі реального часу, а створення візуальної «підкладки» для відображення тестових запитань створюється лише один раз, тому потрібно вибрати бібліотеку, продуктивну при оновленні даних і йде добре з реалізацією маршрутизації клієнтів.

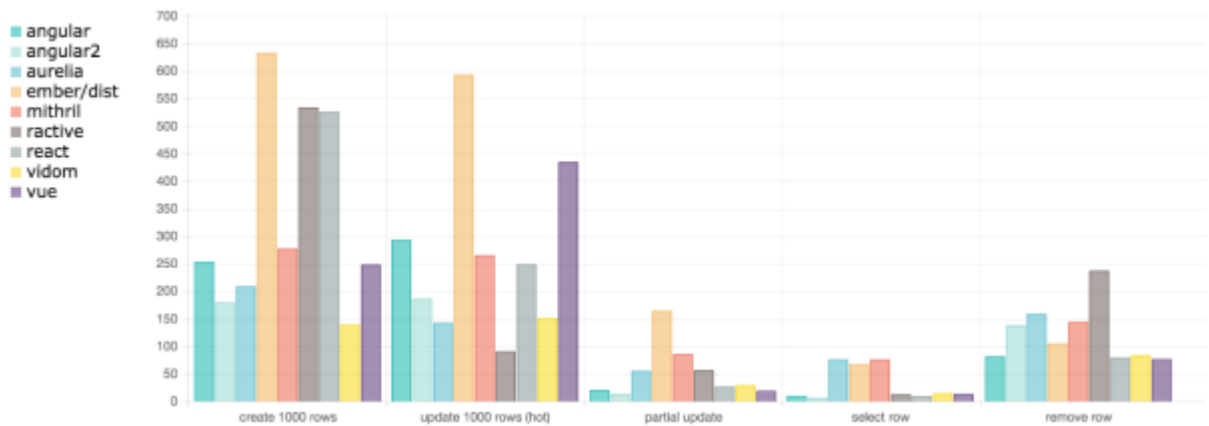


Рисунок 2.1 – Гістограма продуктивності фреймворку JS (мс)

React — це бібліотека JavaScript для створення інтерфейсів користувача. React дозволяє описувати елементи, тобто є декларативним. Використовуючи цю технологію, можна без проблем створювати інтерактивні інтерфейси користувача.

Розробник може розробляти прості уявлення для кожного стану майбутнього веб-додатка, а React може ефективно оновлювати та повторно відобразити лише ті компоненти, які вплинули на зміну даних. Декларативні уявлення роблять код більш передбачуваним для виконання, а також таким, що легше налаштовувати.

Розробник також може створювати інкапсульовані компоненти, які керують своїм власним станом, потім з'єднувати їх для подальшого використання та створювати складні інтерфейси користувача. Оскільки логіка компонентів написана на JavaScript, а не на мові шаблонів, програміст може легко передати досить великий набір даних зі складною структурою по всьому додатку, зберігаючи стан поза межами DOM [2].

Завдяки використанню цього фреймворка стає можливим завантажувати на клієнта (наприклад, у WEB-браузер) усі можливі «Перегляди» відразу. Тобто для кожної дії користувача в WEB-клієнті є відповідне графічне WEB-подання, і з сервера потрібно завантажити лише дані, які були змінені (наприклад, деякі числові дані в моніторингу, або будь-яку іншу інформацію в зручну форму,

наприклад, JSON). Завдяки такому підходу в WEB-клієнт під час початкового завантаження сторінки завантажується відносно великий файл (близько 200-300 КБ для великих WEB-сервісів), однак при наступних запитах у поточному сеансі завантажується з 1 Б. до ~ 2-3 КБ (залежно від розміру змінених даних), при цьому на екрані змінюється лише та частина, до якої було застосовані які-небудь зміни. При даному підході час на завантаження необхідного перегляду та Інтернет-ресурсів користувача стає значно меншим. В результаті спостерігається висока швидкість роботи WEB-сервісу, покращується відгук інтерфейсу, прискорюється перехід між сторінками за рахунок маршрутизації на стороні клієнта, а шаблонування інтерфейсу значно спрощує розробку клієнтської частини WEB-додатка.

React побудований на концепції компонентів. Він відрізняється від таких фреймворків, як Angular або Ember, які використовують двостороннє прив'язування даних для оновлення HTML сторінки. На думку багатьох розробників front-end [3], React легше вивчити, ніж Angular або Ember — він набагато менший і добре працює з jQuery та іншими фреймворками. Даний фреймворк є надзвичайно швидким, оскільки використовує віртуальний DOM і оновлює лише змінені частини сторінки (доступ до DOM все ще є повільною частиною сучасних WEB-додатків, тому бібліотека отримує перевагу в продуктивності, оптимізуючи її).

Маршрутизація на стороні клієнта в цьому випадку може бути реалізована за допомогою бібліотеки react-router [4]. Ця бібліотека дозволяє асоціювати клієнтські маршрути з компонентами React, тому всі можливі стани WEB-інтерфейсу будуть міститися у файлі JavaScript проекту, де оголошуються маршрути (у точці входу програми, згідно з Рис. 2.2).

```

const routes = (
  <Router history={browserHistory}>
    <Redirect from="/" to="/login" />
    <Route path="/" component={MainComponent}>
      <Route path="/login" component={LoginComponent} />
      <Route path="/registration" component={RegistrationComponent} />
      <Route path="/reservation" component={ReservationComponent} onEnter={requireAuth} />
      <Route path="/parking" onEnter={requireAuth}>
        <Route path="auto" component={ParkingComponent} onEnter={autoReserve} />
        <Route path="manual" component={ParkingComponent} />
      </Route>
    </Route>
  </Router>
);

ReactDOM.render(routes, document.getElementById('root'));

```

Рисунок 2.2 – Код JS-файлу точки входу програми, що містить маршрутизацію клієнта

Для малювання діаграм або складних креслень у WEB-інтерфейсі може знадобитися бібліотека D3. D3.js (або просто D3) — це бібліотека JavaScript для маніпуляції та візуалізації даних. Вона надає зручні утиліти для обробки та завантаження масивів даних та створення елементів DOM. Назва D3 означає документ, керований даними.

Це бібліотека JavaScript, орієнтована на роботу з даними та їх візуальне представлення для WEB-додатків, включаючи завантаження даних, візуалізацію в реальному часі та багато інших функцій [5, 6].

Оскільки більшість браузерів, які використовуються сьогодні, не підтримують сучасні стандарти EcmaScript [7], браузери повинні використовувати старий стандарт EcmaScript 5, який підтримується всіма сучасними браузерами, включаючи Internet Explorer 11 [8], що дозволяє значно розширити можливості за допомогою розробленого WEB-інтерфейсу. Однак, відмовляючись від новітніх стандартів мови програмування, розробник також відмовляється від можливості використання нового синтаксичного цукру та деяких оптимізаційних рішень нових стандартів. Для цього необхідно перевести JavaScript новішого стандарту [7] на JavaScript стандарту EcmaScript 5, з цим завданням справляється Babel (компілятор JavaScript, його документація є на сайті [9]).

В результаті виникає досить багато залежностей, які потрібно імпортувати в більшість файлів, і в міру розвитку програми код зростає і з'являється багато

файлів. Щоб зібрати файли JavaScript і файли стилів в один файл JavaScript і файл стилів відповідно, знадобиться система для побудови коду і пакетів у WEB-інтерфейсі, webpack, документація якого знаходиться на сайті [10].

2.3 Бібліотеки CSS

Для того, щоб визначити стиль HTML документів використовується мова CSS. Вона є примітивною і неповною мовою програмування. Створення функції, повторне використання визначення або використання в ній успадкування стає надзвичайно складним процесом. Підтримання коду CSS стає великою проблемою для великих проектів або складних систем. Проте WEB-технології швидко розвиваються, впроваджуються нові специфікації як в HTML, так і в CSS. Веб-браузери використовують ці специфікації, але залишають свої префікси, специфічні для виробника. У деяких випадках (наприклад, фонові градієнти) програмування за допомогою спеціальних префіксів постачальників стає тягарем. Щоб досягти однакового результату, потрібно додати всілякі префікси різних версій браузера.

Для того, щоб покращити код CSS програмісти використовують різні підходи. Наприклад, визначення стилів розбивають на невеликі файли, а вже потім імпортують їх в один великий головний файл.

Такий підхід допоміг розділити стилі на компоненти, але не вирішив проблеми повторення коду і не полегшив його обслуговування. Інший підхід полягав у спробі ввести об'єктно-орієнтоване програмування в CSS. У цьому випадку до елемента було застосовано два або більше визначення класів. Кожен клас додав до цього елемента один тип стилю. Створення кількох класів збільшило можливість успадкування коду, але зменшило здатність ефективно підтримувати код.

Препроцесори, у свою чергу, допомагають писати масштабований і підтримуваний код CSS. Використовуючи препроцесор, програміст може легко підвищити свою продуктивність і зменшити кількість коду в своєму проекті.

Таким чином, для комфортного програмування стилів для WEB клієнта розробнику потрібен препроцесор CSS. Препроцесори CSS розширюють використання CSS за допомогою плагінів, операторів, інтерполяцій, функцій, міксинів та інших корисних інструментів. Найвідомішими препроцесорами CSS є SASS [11], LESS [12] та Stylus [13].

Як і будь-яка мова програмування, препроцесори мають різний синтаксис, але вони дуже схожі один на одного. Усі препроцесори підтримують «класичне» програмування CSS, а їх синтаксис загалом схожий на CSS.

Давайте подивимося, як синтаксис препроцесора CSS відрізняється на прикладі міксинів (Рис. 2.3). Міксини – це набір умов, які формуються за деякими параметрами або статичними правилами. За допомогою них ви можете створювати щось на зразок функцій і повторно використовувати правила стилю в інших місцях вашого коду. Як ви можете бачити на малюнку 3, найбільш знайомий синтаксис, схожий на CSS, - це SASS і LESS, але SASS має більш конкретний, детальний опис використання міксинів (@mixin, @include), ніж дубльований LESS (символ "."). клас селектора, який починається з символу «.», тому розробка буде використовувати препроцесор SASS CSS [14].

```
Sass
@mixin bordered($width) {
  border: $width solid #ddd;

  &:hover {
    border-color: #999;
  }
}

h1 {
  @include bordered(5px);
}

Less
.bordered (@width) {
  border: @width solid #ddd;

  &:hover {
    border-color: #999;
  }
}

h1 {
  .bordered(5px);
}

Stylus
bordered(w)
  border: n solid #ddd
  &:hover
    border-color: #999

h1
  bordered(5px)

CSS
h1 { border: 5px solid #ddd; }
h1:hover { border-color: #999; }
```

Рисунок 2.3 – Опис міксинів у різних препроцесорах та аналог в CSS

Для того, щоб написати крос-браузерний CSS-код більш ефективно необхідно використати утиліту, яка автоматично додає префікси до правил стилів. Для прикладу Google рекомендує автопрефікс [15] для обробки postCSS. Дана бібліотека дозволяє писати стилі без специфічних для виробника префіксів (Рис. 2.4 і 2.5), що і вирішує проблему написання кросбраузерних стилів.

```
:fullscreen a {  
  display: flex  
}
```

Рисунок 2.4 – Опис правила CSS без префіксів постачальника

```
:-webkit-full-screen a {  
  display: -webkit-box;  
  display: flex  
}  
:-moz-full-screen a {  
  display: flex  
}  
:-ms-fullscreen a {  
  display: -ms-flexbox;  
  display: flex  
}  
:fullscreen a {  
  display: -webkit-box;  
  display: -ms-flexbox;  
  display: flex  
}
```

Рисунок 2.5 – Опис правил CSS з префіксами постачальника, після запуску autoprefixer

2.4 Інтеграція стека технологій

Для ефективної інтеграції великої кількості залежностей у проект потрібен менеджер пакетів. У разі відсутності менеджера пакетів усі залежності необхідно завантажувати вручну з джерел Інтернету або залежності потрібно завантажувати в систему управління кодом, що збільшить розмір проекту

розробки в декілька разів і негативно вплине на швидкість завантаження, встановлення проекту та ресурси сервера контролю джерел. Найвідомішим менеджером пакетів для бібліотек JavaScript є `npm`, який, у свою чергу, є найбільшим у світі реєстром програмного забезпечення. Від `npm` немає необхідності додавати файли залежностей до проекту. Замість цього достатньо додати файл під назвою `"package.json"` (Додаток А) до кореня проекту, який містить зіставлення між іменами бібліотек JavaScript та їх версіями. Таким чином, проект завжди матиме необхідні версії пакетів, і ви зможете завантажити їх перед запуском або створенням проекту за допомогою простої команди в консолі встановлення `npm` (розташованої в корені проекту).

Залежності проекту, у свою чергу, можуть містити власні залежності, які будуть завантажені під час команди `«npm install»`. Це стане причиною того, що кодова база проекту буде містити значну кількість коду, більшість з якого не можна стиснути, а деякі частини коду можуть взагалі не бути використані в проекті, проте займати додатковий простір, що є причиною завантаження додаткових даних у браузер під час завантаження сторінки та розміщення даних в оперативній пам'яті, що може сповільнити WEB-інтерфейс. Щоб уникнути цих проблем, вам слід видалити невикористаний код і стиснути його під час складання, але краще робити це автоматично. Це можна зробити у разі налаштування збірки `webpack` [10], а саме, додавши `UglifyJsPlugin` до файлу конфігурації `webpack`, який мінімізує та вимикає JavaScript. Необхідним є також додати `DedupePlugin`, який шукає ідентичні файли в проекті та виключає їх з остаточної збірки. Невикористані файли та бібліотеки `webpack` не включаються у вихідні дані збірки за замовчуванням.

Щоб автоматично використовувати такі технології, як препроцесори CSS, обробка `postCSS` (автопрефікс), сучасні стандарти `Ecmascript` [7], необхідно також додати так звані «завантажувачі» до конфігураційного файлу системи збірки `webpack`.

Такі залежності, як babel-loader, sass-loader, postcss-loader і autoprefixer, що описані у файлі “package.json”, дозволяють програмісту писати сучасний код JavaScript. Іншою важливою перевагою є використання синтаксису препроцесора SASS CSS, замість написання специфічних для постачальників префіксів, оскільки переклад мов і додавання префіксів відбувається автоматично при компіляції.

2.5 Опис середовища та системи

Редактори коду – це незамінний інструмент для всіх, хто хоче краще налаштувати робоче середовище або глибше проникнути в веб-дизайн та розробку. Вони значно полегшують роботу у порівнянні зі стандартними текстовими редакторами.

Наразі існує безліч варіантів і це значно ускладнює вибір правильного інструменту. Провівши поглиблені тести деяких найпопулярніших редакторів коду, середовищем розробки було обрано Sublime Text для виконання даної дипломної роботи.

Sublime можна завантажити з офіційного сайту. Версія 3 зараз доступна в бета-версії як для OS X, так і для Windows та Ubuntu. Існує портативна версія, яку можна легко запустити із USB-накопичувача.

Ви можете завантажити Sublime Text безкоштовно. Після закінчення ознайомлювального періоду ви отримаєте ліцензію на подальше використання. Після вибору бажаної версії завантаження виконується швидко. Розмір інсталяційного архіву становить 8 Мбайт, а після встановлення програма займає лише 22 Мбайт.

Після встановлення настав час першого запуску. У разі успішного встановлення, буде відкрито наступний екран:

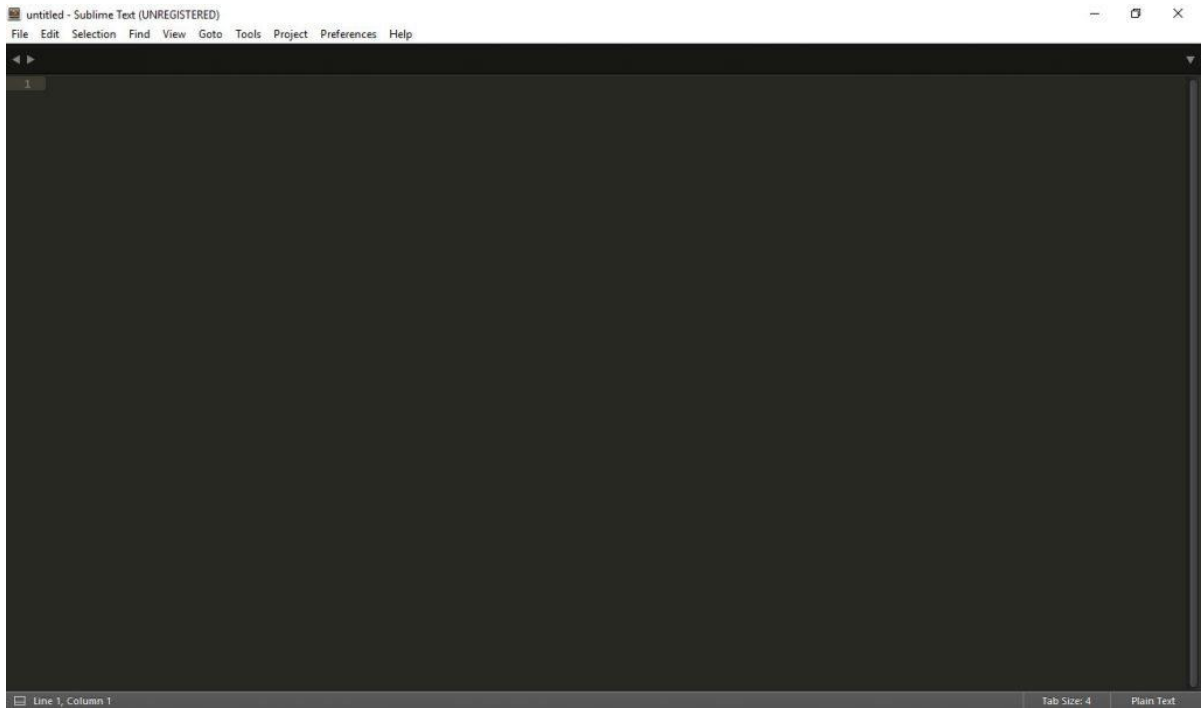


Рисунок 2.6 – Запуск Sublime

Особливості та переваги Sublime Text:

- Автозаповнення, підсвічування синтаксису, можливість згортання коду;
- Із загального дизайну Sublime Text є дуже структурованим та логічним;
- Sublime має вбудовану підтримку кількох десятків мов програмування та відповідним чином розфарбовує елементи коду. Редактор коду навіть попередить вас про помилки парсингу у режимі реального часу.

На додаток до цього, Sublime поставляється з автозаповненням коду, у тому числі для змінних, створених користувачем, та згортанням коду. Останній робить великі фрагменти коду компактнішими для перегляду, приховуючи їх частини.

Sublime Text можна повністю налаштувати за допомогою плагінів. Швидко інтегрований диспетчер пакетів дозволяє користувачам знаходити, встановлювати, оновлювати та видаляти плагіни прямо з редактора – зазвичай без перезавантаження.

Користувачі можуть завантажувати пакети з Github та BitBucket та PackageControl.io. Останній є власним репозиторієм Sublime, де публікується велика кількість плагінів для розширення його можливостей.

Усього доступно більше 3800 пакетів. З їхньою допомогою можна перетворити Sublime з простого редактора коду на повноцінну IDE та побудувати саме той тип середовища розробки, який потрібен.

Наприклад, можна додати можливість синхронізації файлів на віддаленому FTP-сервері, перегляд коду, написаного на Markdown, і розширені елементи управління на бічній панелі. Як і каталог плагінів WordPress, пакети містять велику інформацію про їх функції, встановлення та використання, включаючи скріншоти.

Єдиним недоліком є те, що ця інформація недоступна всередині самого Sublime Text, лише на сайті. З іншого боку, установка через Package Control шалено швидка, і є можливість контролювати налаштування для всіх пакетів у Preferences > Package Settings.

Окрім 22 встановлених тем, які Sublime пропонує з за замовчуванням, є також теми, які доступні як додаткові пакети. Кожен із них дозволяє користувачам змінювати колірну схему свого коду одним натисканням кнопки.

Щось, що стає очевидним під час використання Sublime, - це швидкість роботи програми. Редактор коду, створений на C/C++ та Python, дуже легкий. Після вибор даного редактору в меню «Пуск» Windows він завантажується фактично миттєво.

Ця різниця також помітна під час роботи з великими файлами. Sublime Text може виконувати набагато складнішу роботу, не стикаючись із проблемами.

Стабільність – одна з найбільш вагомих причин, за якої люди віддають перевагу Sublime Text іншим редакторам.

Ще одна визначна пам'ятка Sublime – потужна функція пошуку. Його звичайна функція пошуку та заміни може шукати регулярні висловлювання, здійснювати пошук з урахуванням регістру чи шукати цілі слова. Крім того, він

дозволяє додавати результати пошуку до буфера для подальшого повторного використання.

У даному редакторі наявна функція Goto. Вибравши її, Sublime відкриває панель пошуку. Коли ви починаєте вводити ім'я будь-якого файлу або каталогу у своєму проєкті або активній папці, редактор миттєво відображає параметри можливі варіанти.

Найприємнішим є те, що коли ви переміщуєтеся вниз по списку, Sublime автоматично відкриває виділений файл у фоновому режимі, щоб ви могли побачити чи правильний він. Коли ви його виявили, клацання або натискання Enter відкриє файл у новій вкладці, тому ви можете одразу розпочати роботу з ним.

Зрештою, є палітра команд. Кожен, хто погано запам'ятовує комбінації клавіш, може відкрити цю колекцію доступних команд, використовувати нечіткий пошук, щоб знайти і виконати те, що необхідно.

Одна з найцікавіших особливостей редактора Sublime – можливість використати кілька курсорів. Фактично це означає, що ви можете редагувати те саме одночасно в декількох місцях.

Наприклад, якщо потрібно змінити ім'я однієї змінної у всьому файлі, ви можете вибрати її один раз, зробити так, щоб Sublime помітив решту входження, а потім ввести нове ім'я лише один раз. Це навіть зручніше, ніж використовувати пошук та заміну.

Цю функцію можна використовувати для переформатування кількох рядків коду одночасно. Корисним прикладом цього є створення впорядкованого чи неупорядкованого списку HTML. Уявіть, що ви пишете оператори `` та `` відразу для всіх елементів списку, а не копіюєте та вставляєте їх.

Для імітації серверної частини програми, програмісти часто використовують програму MAMP.



Рисунок 2.7 – Сервер MAMP

MAMP – Mac OS X, Apache, MySQL та PHP. MAMP – це варіант програмного пакету LAMP, який можна встановити в Mac OS X. Його можна використовувати для запуску реального веб-сервера з Mac, але найчастіше він використовується для веб-розробки та локального тестування.

MAMP також включає MySQL і PHP. MySQL і PHP – спільні технології з відкритим вихідним кодом. Вони використовуються для створення сайтів, що є динамічними. MySQL – популярна СУБД, а PHP – мова веб-сценаріїв. Веб-сторінки, що містять PHP-код, можуть отримати доступ до даних з бази даних MySQL і завантажувати динамічний контент надзвичайно швидко. Встановивши PHP та MySQL локально, розробник може створити та протестувати динамічний сайт на своєму Mac перед публікацією його в Інтернеті.

MAMP також включає графічний інтерфейс користувача, який можна використовувати для управління локальним веб-сервером.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ЕЛЕКТРОНІКИ

3.1 Розробка моделі інформаційної системи

Першочерговим завданням є створення початкової (головної) сторінки, що складається з таких частин:

- Header;
- Aside;
- Main;
- Footer.

Для підключення до бази даних був використаний код (Додаток В).

У header знаходиться логотип компанії і три іконки: бажання, корзина і профіль. Код header винесено в окремий php файл для більш зручного використання багатосторінкового сайту:

```
<header>
  <div class="container">
    <div class="row">
      <div class="col-12 d-flex justify-content-between align-items-center">
        <a href="index.php"></a>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
          <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="toggle navigation">
              <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
              <ul class="navbar-nav">
                <li class="hp nav-item">
                  <a class="nav-link" href="index.php">Головна</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link" href="Vidguk.php">Відгуки</a>
                </li>
              </ul>
            </div>
          </div>
        </nav>
        <ul class="d-flex m-3 p-0">
          <li class="mx-2"><a href="cart.php"></a></li>
          <li class="mx-2"><a href="wishes.php"></a></li>
          <li class="mx-2"><a data-bs-toggle="modal" data-bs-target="#exampleModal"></a></li>
        </ul>
      </div>
    </div>
  </header>
  <?php include('modal.php') ?>
```

В усі головні сторінки сайту даний файл підключений наступний чином:

```
<?php include('header.php') ?>
```

У aside зберігатися список товарів, що розбиті по категоріям, який завантажується через РНР з бази даних. Даний файл було також винесено в окремий php файл, що надає можливість швидкої зміни всього вмісту відразу у всіх файлах. Код наведено нижче:


```

<div class="col-12 aside scrollmenu p-0">
  <ul class="px-0 d-flex flex-row flex-md-column m-0">
    <li class="py-2 px-4"><a href="index.php">Популярне</a></li>
    <li class="py-2 px-4"><a href="comp.php">Комп'ютери</a></li>
    <li class="py-2 px-4"><a href="tech.php">Побутова техніка</a></li>
    <li class="py-2 px-4"><a href="phone.php">Телефони</a></li>
    <li class="py-2 px-4"><a href="tele.php">Телевізори</a></li>
  </ul>
</div>

```

Даний файл підключено наступним чином:

```
<?php include('aside.php') ?>
```

В main відображається обрана категорія товарів. Усі товари завантажуються з бази даних. Код даного функціоналу наведено нижче:

```

<?php
    $categories = get_categories($link, "popular");
    ?>
    <?php foreach ($categories as $popular): ?>
    <div class="col-lg-4 col-6 p-2">
      <div class="wrapper col-12 p-2 d-flex flex-column align-items-center justify-content-between">
        <img <?=$popular["fotoUrl"] ?> alt="comp" class="img-fluid">
        <h4><?=$popular["name"] ?></h4>
        <p><?=$popular["description"] ?></p>
        <div class="price my-2">Цена: <span><?=$popular["price"] ?></span></div>
        <div class="d-flex">
          <form method="POST" action="add to wish.php" class="d-flex">
            <input type="text" name="id" value='<?=$popular["id"] ?>' class="d-none">
            <input type="text" name="category" value='popular' class="d-none">
            <button name="heart" class="btn btn-primary mx-1">В избранное</button>
          </form>
          <form method="POST" action="add to cart.php" class="d-flex">
            <input type="text" name="id" value='<?=$popular["id"] ?>' class="d-none">
            <input type="text" name="category" value='popular' class="d-none">
            <button type="submit" name="cart" class="btn btn-success mx-1">В корзину</button>
          </form>
        </div>
      </div>
    </div>
    </div>
    <?php endforeach; ?>

```

Сама функція вибору товару виглядає наступним чином:

```

<?php
function get_categories($link, $cat) {
    $sql = "SELECT * FROM $cat";
    $result = mysqli_query($link, $sql);
    $categories = mysqli_fetch_all($result, MYSQLI_ASSOC);
    return $categories;
}
?>

```

У footer відображається логотип компанії, контакти і можливі способи зв'язку. Footer також винесено в окремий php файл:

```

<footer>
  <div class="container">
    <div class="row">
      <div class="col-12 d-flex justify-content-between align-items-center">
        
        <h4>€#169; 2022</h4>
        <ul class="p-0 m-3 d-flex">
          <li class="mx-2"><a target="_blank" href="https://www.instagram.com"></a></li>
          <li class="mx-2"><a target="_blank" href="https://www.facebook.com"></a></li>
          <li class="mx-2"><a target="_blank" href="https://www.youtube.com"></a></li>
        </ul>
      </div>
    </div>
  </div>
</footer>

```

До всіх файлів зі сторінками підключений наступним чином:

```
<?php include('footer.php') ?>
```

З головної сторінки є можливість перейти на наступні розділи:

- Телевізори;
- Телефони;
- Побутова техніка;
- Комп'ютери;
- Кошик;
- Обране;
- Профіль.

Для забезпечення швидкого контакту з адміністраторами інтернет-магазину було реалізовано перехід на різні соціальні мережі.

З кожною сторінки можна повернутися назад на головну, натиснувши на логотип компанії.

Основаючись на вище сказаному, було побудовано наступну структуру додатку (Рис. 3.1)

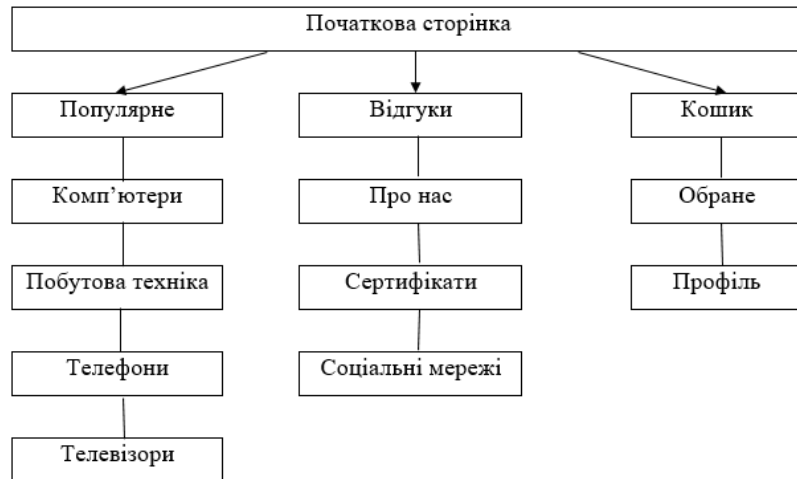


Рисунок 3.1 – Структура додатку

Для візуалізації процесів та роботи систем розробниками програмного забезпечення використовується уніфікована мова моделювання UML (Unified Modeling Language). Саме для того, щоб структура додатку «Electron» була зрозуміла кожному, було створено діаграму взаємодії користувачів, яка виглядає наступним чином (Рис. 3.2).

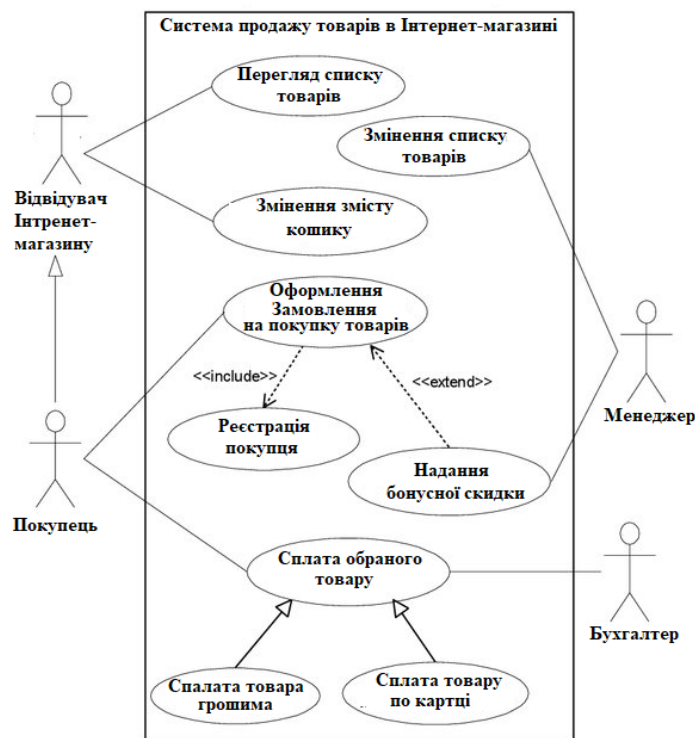


Рисунок 3.2 – Діаграма UML

3.2 Програмна реалізація

У системі керування контентом застосовується СУБД MySQL [3]. База даних складається з основних таблиць:

- Кошик;
- Комп'ютери;
- Побутова техніка;
- Популярне;
- Телефони;
- Обране;
- Телевізори;
- Відгуки;
- Користувачі.

Для роботи була розроблена наступна структура бази даних (Рис. 3.3).

Cart
– id
– fotoUrl
– name
– description
– price

Wish
– id
– fotoUrl
– name
– description
– price

Users
– id
– login
– pass
– surname
– name
– fotourl
– address
– phone

Otziv
– id
– textotziv
– username
– fotourl

Comp
– id

Phone
– id

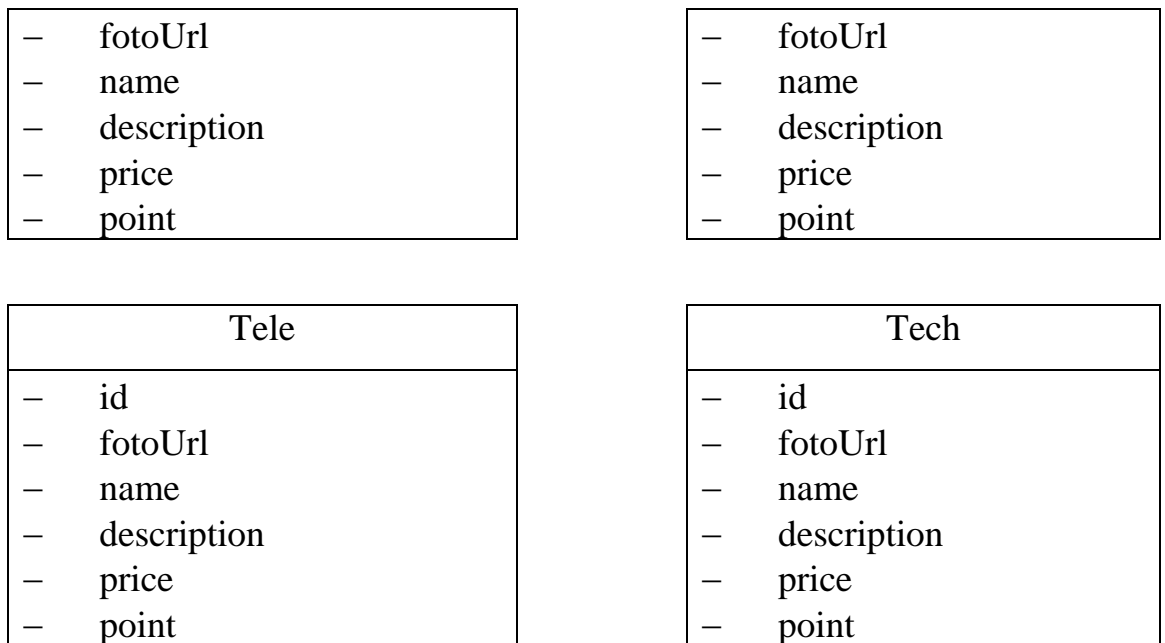


Рисунок 3.3 – Структура бази даних

Всі дані зберігаються на локальному сервері в окремих базах даних (Рис. 3.4).

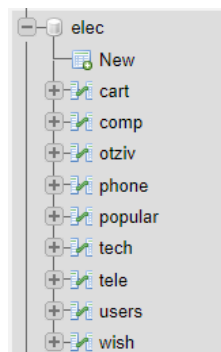


Рисунок 3.4 – Структура БД

Нижче описані всі структури баз даних, які були створені для нормальної роботи сайту (Рис. 3.5-3.13).

+ Options			id	fotoUrl	name	description	price
<input type="checkbox"/>	Edit Copy Delete	34	src="img/tele4.jfif"	Television on wall	Television on wall	30000	
<input type="checkbox"/>	Edit Copy Delete	21	src="img/comp1.jfif"	PC computer	Computer for games	60000	
<input type="checkbox"/>	Edit Copy Delete	45	src="img/elect5.jfif"	Vacuum cleaner	vacuum cleaner	15000	
<input type="checkbox"/>	Edit Copy Delete	13	src="img/mob3.jfif"	IPhone	IPhone last version	80000	

Рисунок 3.5 – Структура БД (кошик)

		id	fotoUrl	name	description	price	point
<input type="checkbox"/>	Edit Copy Delete	21	src="img/comp1.jfif"	PC computer	Computer for games	60000	9
<input type="checkbox"/>	Edit Copy Delete	22	src="img/comp2.jfif"	Compact computer	Compact computer	10000	2
<input type="checkbox"/>	Edit Copy Delete	23	src="img/comp3.jfif"	Stylish computer	Stylish computer	35000	7
<input type="checkbox"/>	Edit Copy Delete	24	src="img/comp4.jfif"	Laptop	Very useful laptop	7000	3

Рисунок 3.6 – Структура БД (Комп'ютери)

		id	textotziv	username	fotourl
<input type="checkbox"/>	Edit Copy Delete	1	Придбала телефон як подарунок для свого чоловіка, ...	@svitlana_m	src="img/p2.jpg"
<input type="checkbox"/>	Edit Copy Delete	2	Дуже вразила якість і ціна, на інших сайтах набага...	@alex_v	src="img/p1.jpg"
<input type="checkbox"/>	Edit Copy Delete	3	Вибрала собі чудовий телевізор на цьому сайті роже...	@kira2012	src="img/p3.jpg"
<input type="checkbox"/>	Edit Copy Delete	4	Придбав комп'ютер про якій мріяв дуже давно, смарт...	@vlad01	src="img/p4.jpg"
<input type="checkbox"/>	Edit Copy Delete	5	Все супер	a89307	src="img/p3.jpg"

Рисунок 3.7 – Структура БД (Відгуки)

		id	fotoUrl	name	description	price	point
<input type="checkbox"/>	Edit Copy Delete	34	src="img/tele4.jfif"	Television on wall	Television on wall	30000	10
<input type="checkbox"/>	Edit Copy Delete	21	src="img/comp1.jfif"	PC computer	Computer for games	60000	8
<input type="checkbox"/>	Edit Copy Delete	45	src="img/elect5.jfif"	Vacuum cleaner	vacuum cleaner	15000	9
<input type="checkbox"/>	Edit Copy Delete	13	src="img/mob3.jfif"	IPhone	IPhone last version	80000	6

Рисунок 3.8 – Структура БД (Популярне)

		id	fotoUrl	name	description	price	point
<input type="checkbox"/>	Edit Copy Delete	11	src="img/mob1.jfif"	Old school phone	Old school phone	1000	2
<input type="checkbox"/>	Edit Copy Delete	12	src="img/mob2.jfif"	Smartphone	Very powerful Smartphone	10000	6
<input type="checkbox"/>	Edit Copy Delete	13	src="img/mob3.jfif"	IPhone	IPhone last version	80000	10
<input type="checkbox"/>	Edit Copy Delete	14	src="img/mob4.jfif"	accessories	accessories for the phone	7500	5
<input type="checkbox"/>	Edit Copy Delete	15	src="img/mob5.jfif"	headphones	Different headphones	3800	4

Рисунок 3.9 – Структура БД (Телефони)

+ Options

				id	fotoUrl	name	description	price			
<input type="checkbox"/>		Edit		Copy		Delete	33	src="img/tele3.jfif"	Smart TV	Smart television	50000
<input type="checkbox"/>		Edit		Copy		Delete	24	src="img/comp4.jfif"	Laptop	Very useful laptop	7000
<input type="checkbox"/>		Edit		Copy		Delete	43	src="img/elect3.jfif"	fridge	Smart fridge	43000
<input type="checkbox"/>		Edit		Copy		Delete	14	src="img/mob4.jfif"	accessories	accessories for the phone	7500

Рисунок 3.10 – Структура БД (Обране)

+ Options

					id	fotoUrl	name	description	price	point		
<input type="checkbox"/>		Edit		Copy		Delete	41	src="img/elect1.jfif"	Electronic microwave	Electronic microwave	10000	10
<input type="checkbox"/>		Edit		Copy		Delete	42	src="img/elect2.jfif"	Electric kettle	Electric kettle	1500	6
<input type="checkbox"/>		Edit		Copy		Delete	43	src="img/elect3.jfif"	fridge	Smart fridge	43000	9
<input type="checkbox"/>		Edit		Copy		Delete	44	src="img/elect4.jfif"	Toothbrush	Electric toothbrush	2500	8
<input type="checkbox"/>		Edit		Copy		Delete	45	src="img/elect5.jfif"	Vacuum cleaner	vacuum cleaner	15000	2

Рисунок 3.11 – Структура БД (Техніка)

+ Options

					id	fotoUrl	name	description	price	point		
<input type="checkbox"/>		Edit		Copy		Delete	31	src="img/tele1.jfif"	Big television	Very powerful television	15000	4
<input type="checkbox"/>		Edit		Copy		Delete	32	src="img/tele2.jfif"	Old school television	Very old television	5000	3
<input type="checkbox"/>		Edit		Copy		Delete	33	src="img/tele3.jfif"	Smart TV	Smart television	50000	2
<input type="checkbox"/>		Edit		Copy		Delete	34	src="img/tele4.jfif"	Television on wall	Television on wall	30000	6

Рисунок 3.12 – Структура БД (Телевізори)

					id	login	password	sername	name	fotourl	adress	phone		
<input type="checkbox"/>		Edit		Copy		Delete	1	levinalexeysergeevich@gmail.com	24138a24b275c5a0d276701552e11066	Іванов	Іван	src="img/p1.jpg"	Харківська7а	1234678
<input type="checkbox"/>		Edit		Copy		Delete	2	levina@gmail.com	24138a24b275c5a0d276701552e11066	Міроненко	Дар'я	src="img/p2.jpg"	Харківська 8а	111111
<input type="checkbox"/>		Edit		Copy		Delete	3	w@gmail.com	49ffc019350692a03e82555eff3cc9b	Q	W	src="img/p4.jpg"	E	1234

Рисунок 3.13 – Структура БД (Користувачі)

Підключення до бази даних відбувається за допомогою функції:

```

<?php
$link = mysqli_connect('localhost','root','root','elec');
if (mysqli_connect_errno())
{
    echo 'Ошибка ('.mysqli_connect_errno().'): '.mysqli_connect_error();
    exit();
}
?>

```

Вибір даних з масиву відбувається за допомогою наступного коду:

```

<?php
    $categories = get_categories($link,"elect");
?>
<?php foreach ($categories as $select): ?>

<?php
function get_categories($link,$cat){
    $sql = "SELECT * FROM $cat";
    $result = mysqli_query($link, $sql);
    $categories = mysqli_fetch_all($result, MYSQLI_ASSOC);
    return $categories;
}
?>

```

Для можливості додавання у кошик була розроблена функція для взаємодії з БД:

```

<?php
include('database.php');
include('function.php');
$id=$_POST['id'];
$cat=$_POST['category'];
echo $id;
add_to_cart($cat,$id);
function add_to_cart($cat,$id){
    global $link;
    $categories = get_categories($link,$cat);
    foreach ($categories as $category){
        $fotoUrl=$category["fotoUrl"];
        $name=$category["name"];
        $description=$category["description"];
        $price=$category["price"];
        if ($category["id"]== $id){
            mysqli_query($link,"INSERT INTO cart (id,fotoUrl,name,description,price)
                VALUES ('$id','$fotoUrl','$name','$description','$price')");
        }
    }
}
header('Location: ' . $_SERVER['HTTP_REFERER']);
?>

```


Схожа функція була розроблена і для обраного:

```
<?php
include('database.php');
include('function.php');
$id=$_POST['id'];
$category=$_POST['category'];
echo $id;
add_to_wish($category,$id);
function add_to_wish($category,$id){
    global $link;
    $categories = get_categories($link,$category);
    foreach ($categories as $category){
        $fotoUrl=$category["fotoUrl"];
        $name=$category["name"];
        $description=$category["description"];
        $price=$category["price"];
        if ($category["id"]== $id){
            mysqli_query($link,"INSERT INTO wish (id,fotoUrl,name,description,price)
            VALUES ('$id','$fotoUrl','$name','$description','$price')");
        }
    }
}
header('Location: ' . $_SERVER['HTTP_REFERER']);
?>
```

Основним компонентом будь-якого веб-сервісу є логотип. Логотип компанії (Рис. 3.14).

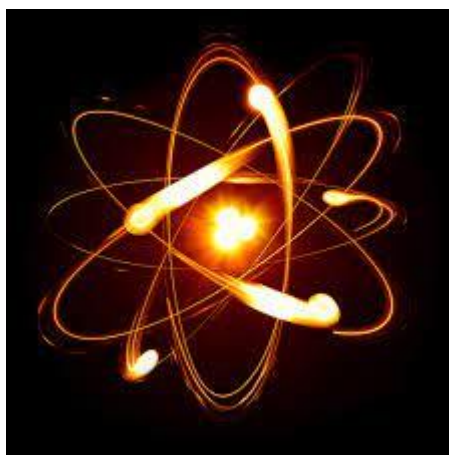


Рисунок 3.14 – Логотип

Всі сторінки містять в собі header, aside та footer. Header виглядає наступним чином (Рис. 3.15):

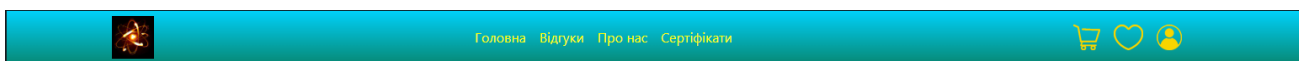


Рис. 3.15 – Header

Aside має вигляд (Рис. 3.16):

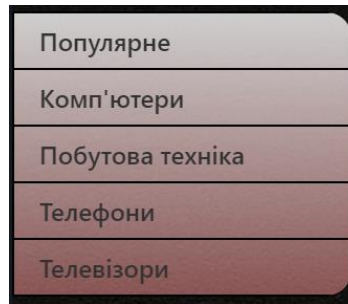


Рисунок 3.16 – Aside

Footer виглядає наступним чином (Рис. 3.17):



Рисунок 3.17 – Footer

3.3 Тестування

Розглянемо приклад реалізації розробленої системи. При введенні адресу магазину відвідувач потрапляє на його головну сторінку (Рис. 3.18). На головній сторінці міститься каталог товарів, контактні дані магазину, а також спеціальні пропозиції. Зовнішній вигляд товарів цієї категорії визначає адміністратор системи.

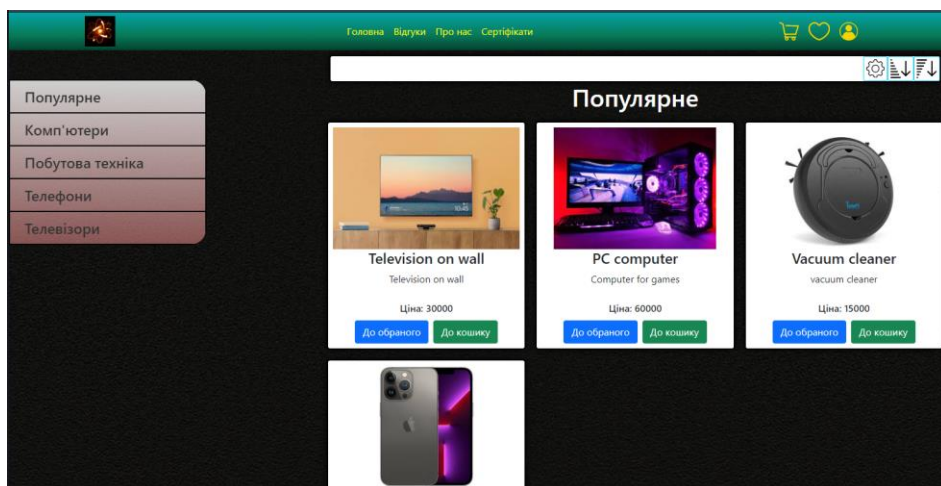


Рисунок 3.18 – Головна сторінка

Усі можливі комп'ютери, що наявні в магазині, відображені на сторінці «Комп'ютери» (Рис. 3.19):

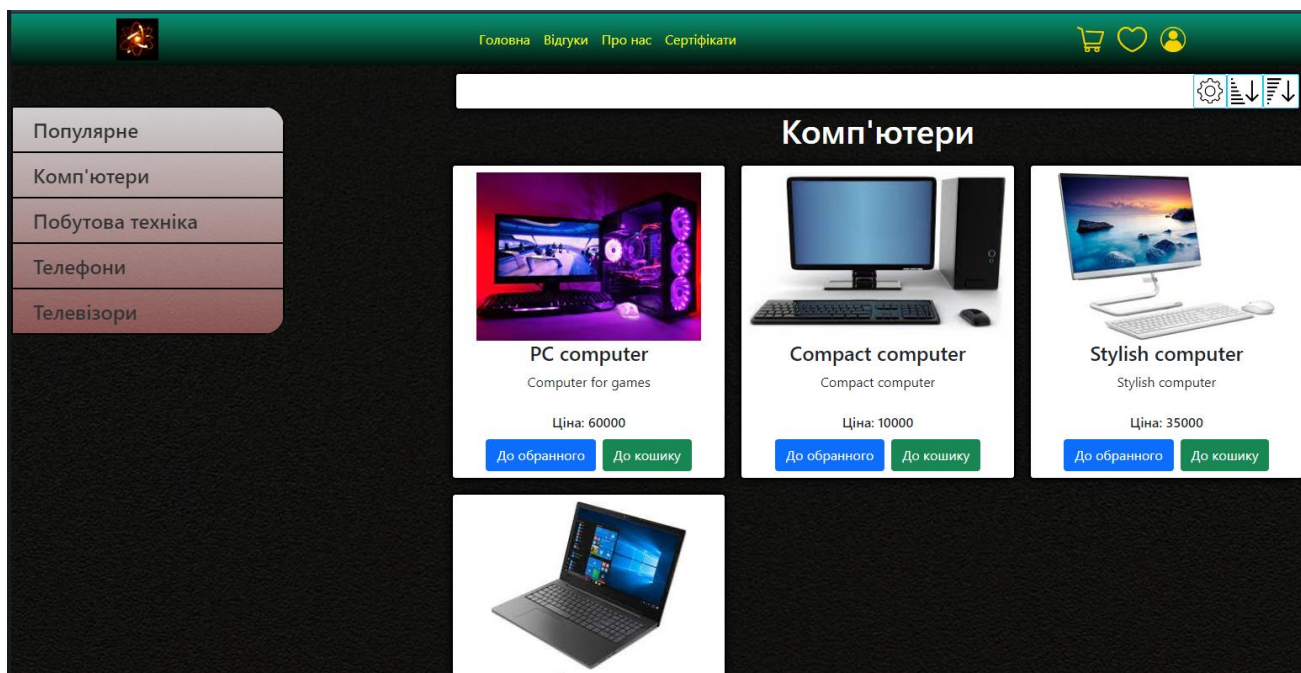


Рисунок 3.19 – Сторінка «Комп'ютери»

Уся можлива техніка, що є в магазині, відображена на сторінці «Побутова техніка» (Рис. 3.20):

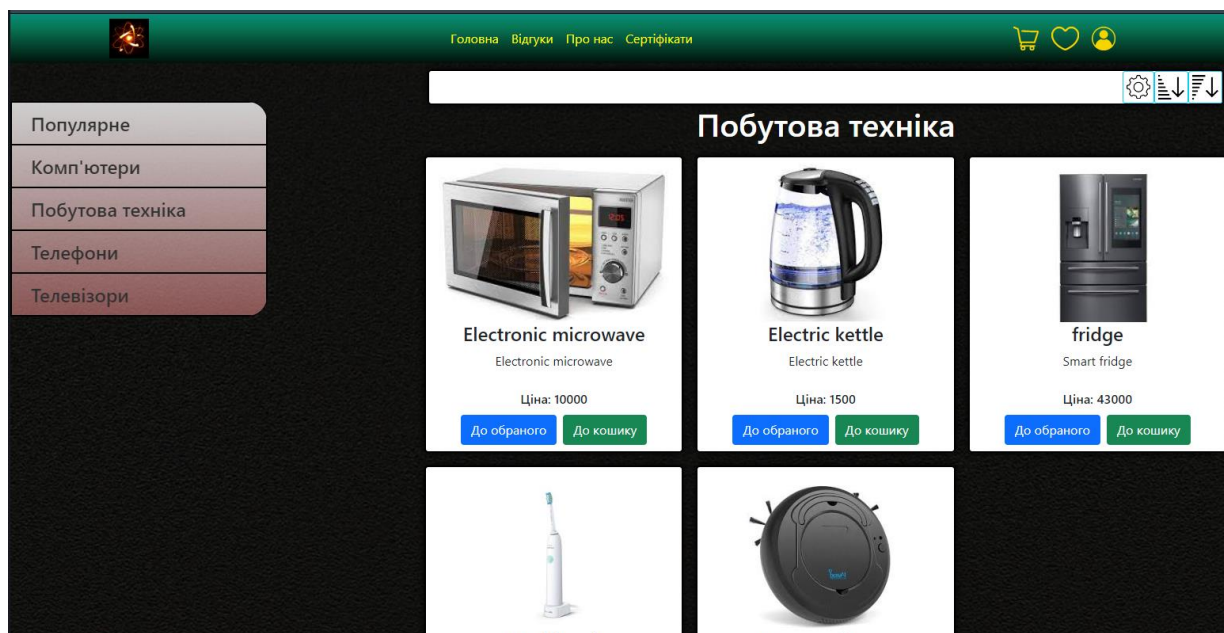


Рисунок 3.20 – Сторінка «Побутова техніка»

Важливим є забезпечення адаптивності сайту для того, щоб користувачі мали можливість заходити на сайт з мобільного телефону чи планшету (Рис. 3.21).

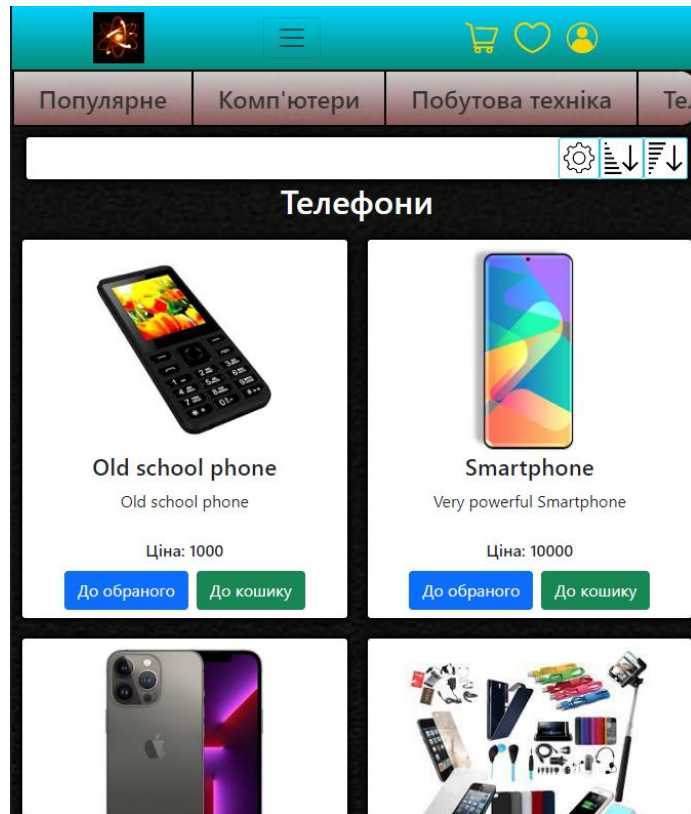


Рисунок 3.21 – Сторінка «Телефони», мобільна версія

Усі телевізори зібрані на сторінці «Телевізори» (Рис. 3.22):

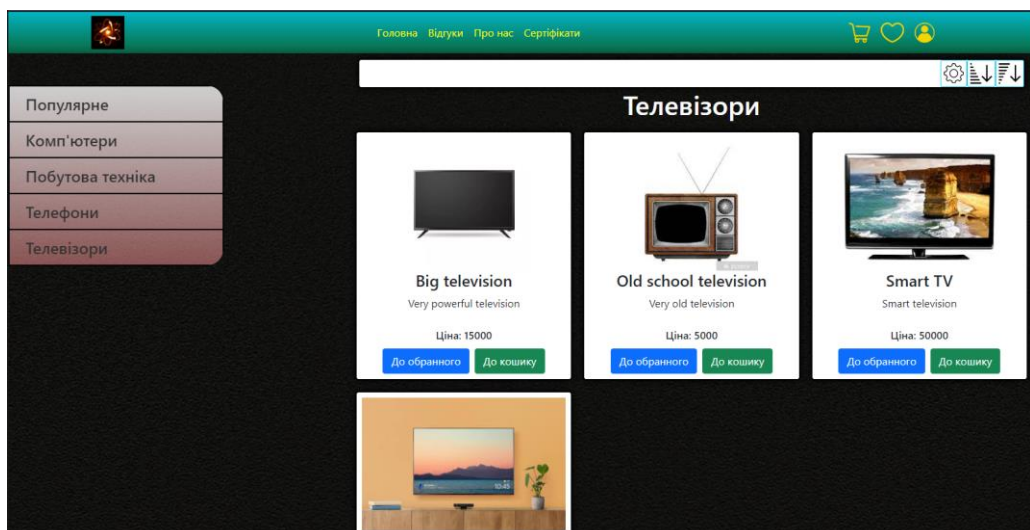


Рисунок 3.22 – Сторінка «Телевізори»

Один із можливих прикладів додавання до кошику (Рис. 3.23):

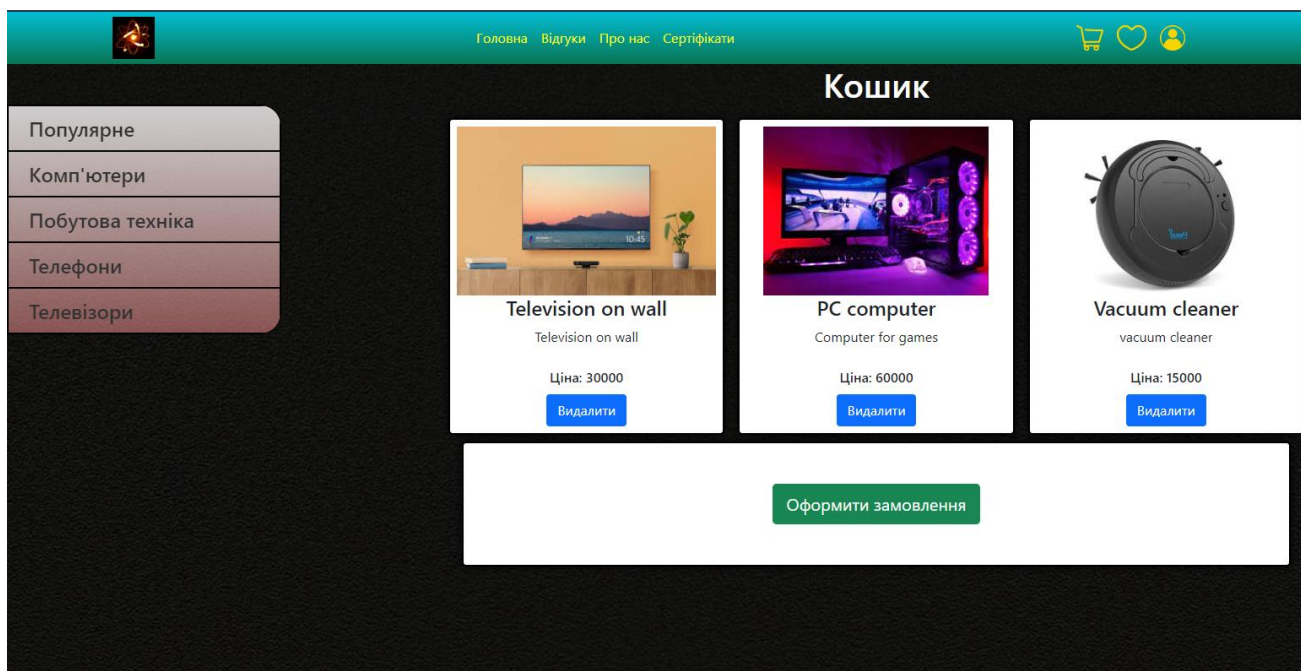


Рисунок 3.23 – Сторінка «Кошик»

Один із можливих прикладів додавання в обране (Рис. 3.24).

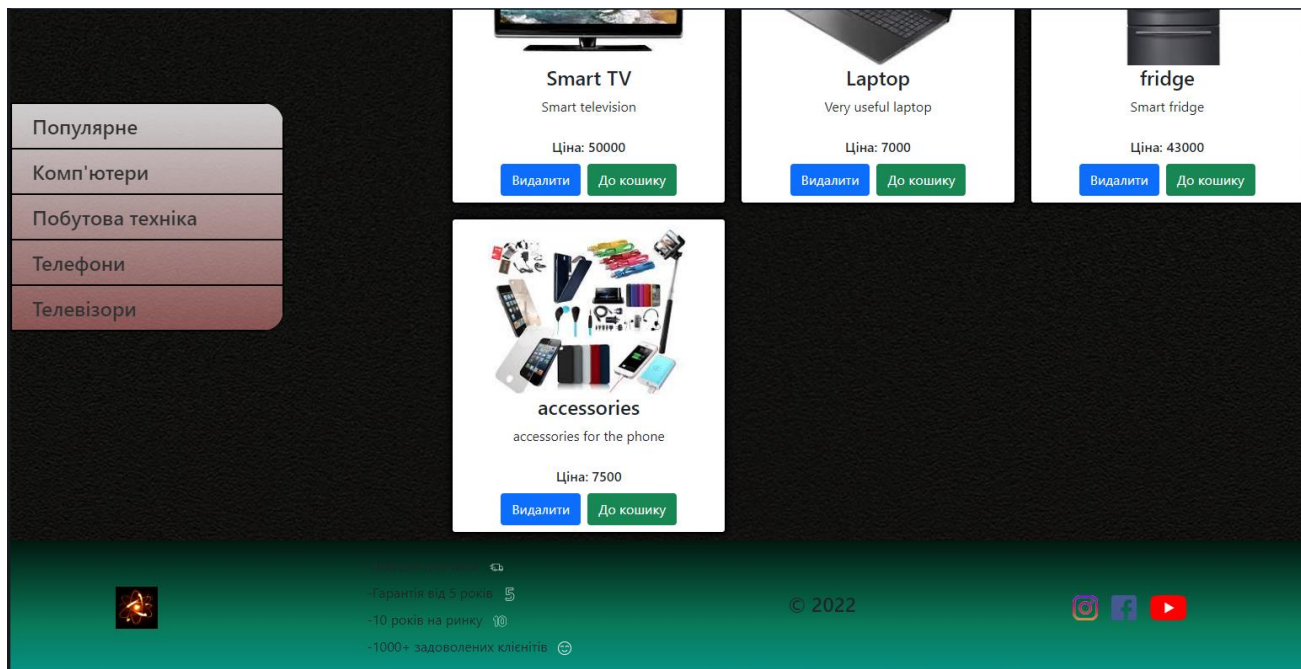


Рисунок 3.24 – Сторінка «Обране»

В обраному можна видалити або додати до кошику. Для входу до магазину було створено спливаюче вікно, яке виглядає наступним чином (Рис. 3.25).

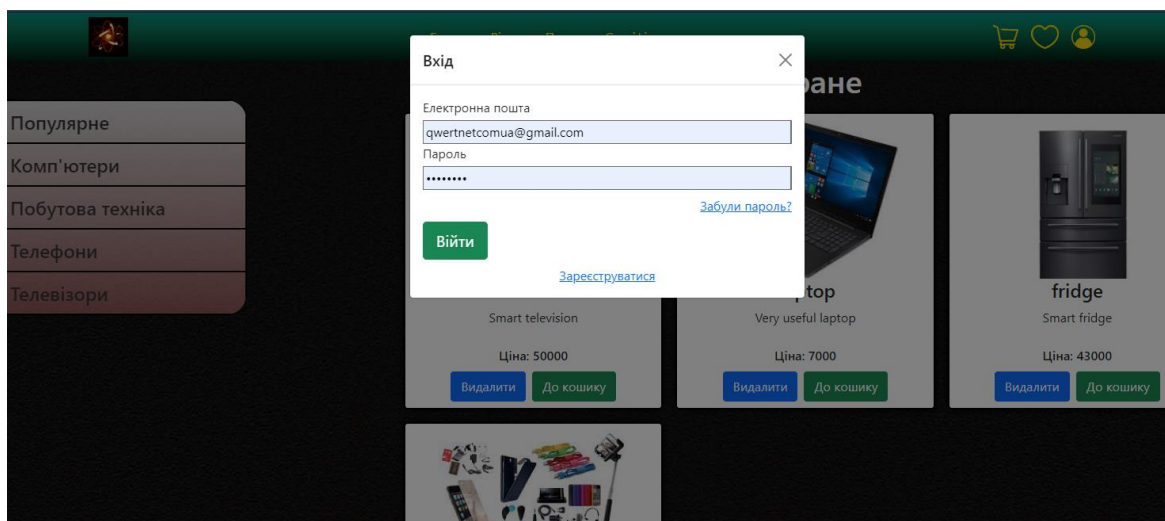


Рисунок 3.25 – Вхід користувача

Для реєстрації користувача була створена окрема сторінка, яка виглядає наступним чином (Рис. 3.26):

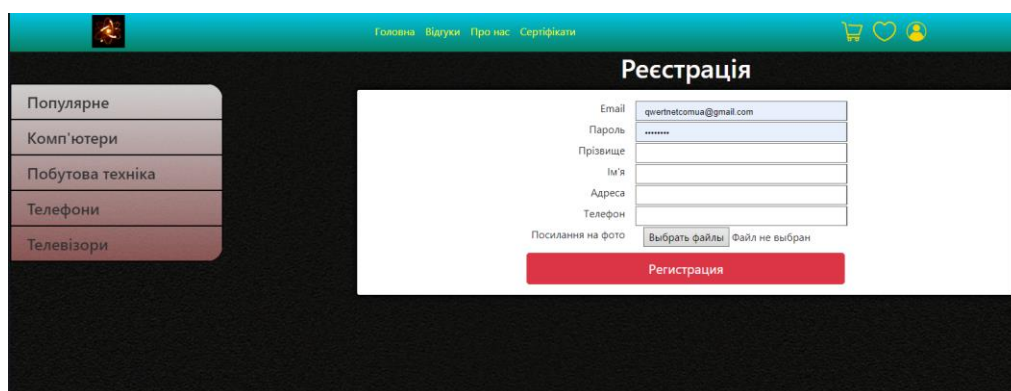


Рисунок 3.26 – Реєстрація користувача

Після реєстрації, користувач може переглянути власні дані (рис. 3.27)



Рисунок 3.27 – Сторінка «Мої дані»

Також для зручного користування пошуку товарі було зроблено сортування за ціною (за зростанням чи спаданням, а також пошук в зазначеному діапазоні, Рис. 3.28).

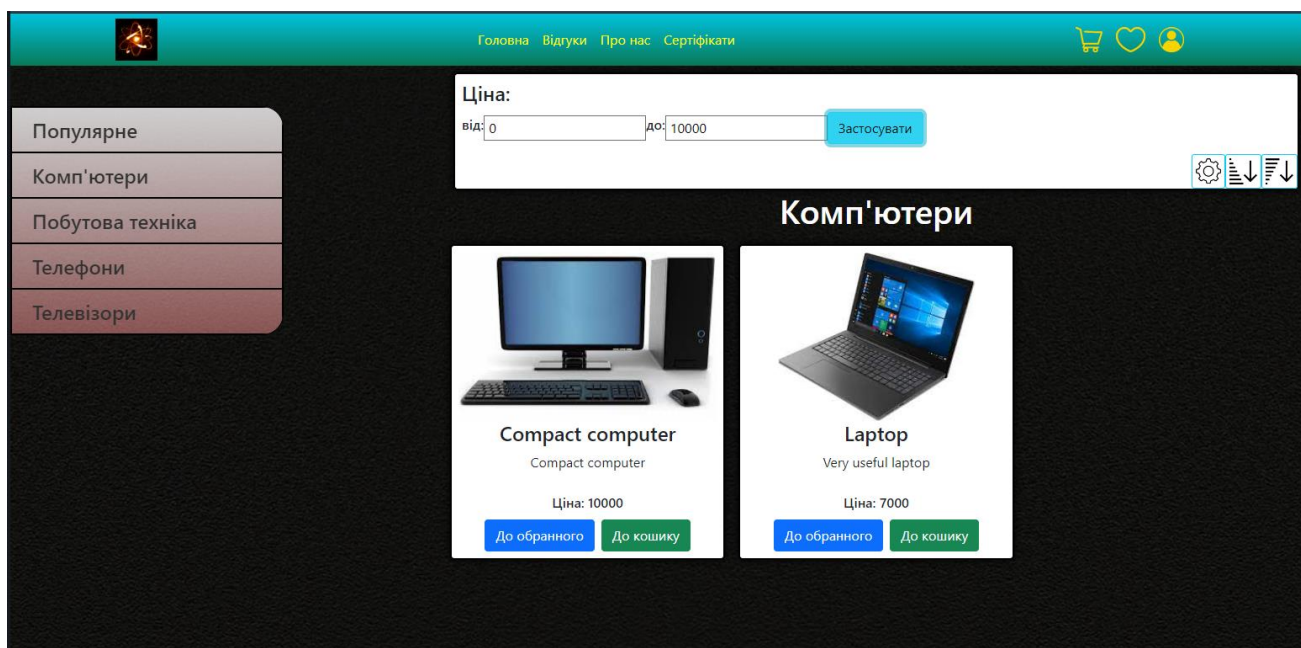


Рисунок 3.28 – Сортування товарів

В результаті роботи вийшов працездатний інтернет магазин, який можна використовувати у комерційній сфері. Надалі розроблений продукт можна доповнити різним функціоналом, додавши до нього різні методи. Використовуючи даний інтернет магазин, підприємство має змогу збільшити власні прибутки, за рахунок інтернет користувачів, які будуть заходити на даний веб-ресурс.

ВИСНОВКИ

Проведено аналіз предметної області щодо побудови інформаційних систем для магазину електроніки з реалізацією алгоритму підбору рекомендованих товарів та визначені основні задачі для досягнення основної мети.

Висвітлено особливості розвитку сучасних веб-сервісів з точки зору проектування дизайну веб-сервісу. Проаналізовано новітні технології розробки веб-сервісів та обрана технологія для програмної реалізації поставленого завдання.

Реалізовано веб-сервіс, який дозволить власнику значно збільшити продажі товарів і послуг, розширити кількість покупців, а також надасть можливість скоротити період часу на весь робочий процес - від надання продукту до його продажу, покращення управління веб-сервісом, зниження витрат на оренду або купівлю торгових приміщень та найму численного персоналу.

На сьогоднішній день існує величезна кількість технологій розробки веб-сервісів, для того щоб розробити якісний веб-сервіс, необхідно використовувати і застосовувати сучасні інструменти розробки.

Цей веб-сервіс розроблено з використанням HTML+CSS+JS та мови програмування PHP та СУБД MySQL. Переважно через те, що вони безкоштовні та за рядом інших критеріїв, ці інструменти є найкращим вибором.

Веб-сервіс розміщено на віртуальному хостингі.

Розроблений інтернет-магазин має перспективи для подальшого розвитку та вдосконалення.

СПИСОК ЛІТЕРАТУРИ

- 1) Розробка інтернет-магазину // Електронна версія на сайті <https://uk.wordpress.org/>
- 2) Загальна інформація про CMS WordPress // Електронна версія на сайті <https://uk.wikipedia.org/wiki/WordPress>
- 3) Загальна інформація про базу даних MySQL // Електронна версія на сайті <https://uk.wikipedia.org/wiki/MySQL>
- 4) Плагін WooCommerce // Електронна версія на <https://woocommerce.com/>
- 5) Загальна інформація про мову програмування Java // Електронна версія на сайті <https://uk.wikipedia.org/wiki/Java>
- 6) Робін Ніксон // Створення динамічних веб-сайтів за допомогою PHP, MySQL, JavaScript, CSS та HTML5, 2016 – 768с.
- 7) Веллінг, Томсон // Розробка веб-додатків за допомогою PHP та MySQL, 2017 – 768 с.
- 8) Брайан, Джейсон // Розробка веб-додатків за допомогою WordPress: WordPress як фреймворк, 2021 – 282 с.
- 9) Фрімен, Робсон // Програмування на JavaScript, 2022 - 457 с.
- 10) Дакетт, Д // HTML та CSS. Розробка та дизайн веб-сайтів / Д. Дакетт, 2011 – 480 с.
- 11) Фрейн Б // Розробка сайтів для будь-яких браузерів і пристроїв, 2017. - 188 с.
- 12) Девід Макфарланд // Нова велика книга CSS, 2017 - 720 с.
- 13) Пілгрім, Марк // Занурення в HTML5 / Марк Пілгрім, 2011 - 304 с.
- 14) Хоган Брайан // Книга веб-програміста: секрети професійної розробки веб-сайтів, 2013 - 512 с.
- 15) Хоган Брайан // HTML5 та CSS3. Веб-розробка за стандартами нового покоління / Брайан Хоган, 2012 – 272 с.

ДОДАТОК

Додаток А(header.php)

```
<header>
  <div class="container">
    <div class="row">
      <div class="col-12 d-flex justify-content-between
align-items-center">
        <a href="index.php"></a>
        <nav class="navbar navbar-expand-lg navbar-light
bg-light">
          <div class="container-fluid">
            <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
              <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarNav">
              <ul class="navbar-nav">
                <li class="hp nav-item">
                  <a class="nav-link"
href="index.php">Головна</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link"
href="Vidguk.php">Відгуки</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link"
href="about.php">Про нас</a>
                </li>
                <li class="hp nav-item">
                  <a class="nav-link"
href="sertificate.php">Сертифікати</a>
                </li>
              </ul>
            </div>
          </div>
        </nav>

        <ul class="d-flex m-3 p-0">
          <li class="mx-2"><a href="cart.php"></a></li>
          <li class="mx-2"><a href="wishes.php"></a></li>
        </ul>
      </div>
    </div>
  </div>
```

```

                <li class="mx-2"><a data-bs-toggle="modal"
data-bs-target="#exampleModal"></a></li>
            </ul>
        </div>
    </div>
</div>

```

```

</div>
</header>

```

```

<?php include('modal.php')?>

```

Додаток Б(footer.php)

```

<footer>
    <div class="container">
        <div class="row">
            <div class="col-12 d-flex justify-content-between
align-items-center">
                

                <ul class="p-0 m-3">
                    <li>-Швидка доставка </li>

                    <li>-Гарантія від 5 років </li>
                    <li>-10 років на ринку </li>

                    <li>-1000+ задоволених клієнтів </li>
                </ul>

                <h4>© 2022</h4>

                <ul class="p-0 m-3 d-flex">
                    <li class="mx-2"><a target="_blank"
href="https://www.instagram.com"><img
alt="insta"></a></li>
                    <li class="mx-2"><a target="_blank"
href="https://www.facebook.com"><img
alt="fb"></a></li>
                    <li class="mx-2"><a target="_blank"
href="https://www.youtube.com"><img
alt="youtube"></a></li>
                </ul>
            </div>
        </div>
    </div>
</footer>

```

Додаток В(registration.php)

```
<?php
require_once 'database.php';
require_once 'function.php';
?>
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ0J3BCsw2P0p/We"
crossorigin="anonymous">
<?php include('css.php') ?>
<title>CLOCKS</title>
</head>
<body>
<?php include('header.php') ?>
<main>
<div class="row m-0">
<div class="col-12 col-md-4 p-0">
<?php include('aside.php') ?>
</div>
<div class="main_section col-12 col-md-8">
<div class="col-12 d-flex flex-column align-items-center">
<h1>Рєєєтгаїя</h1>
<div class="wrapper p-3">
<form method="post" id="form_reg"
action="check_registrate.php">
<div class="d-flex row align-items-center m-0"
id="block-form-registration">
<label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Email</label>
<input type="email" name="reg_email"
id="reg_email" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
<label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Пароль</label>
<input type="password"
name="reg_pass" id="reg_pass" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
<label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Прїзвїице</label>
<input type="text"
name="reg_surname" id="reg_surname" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
<label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Ім'я</label>
<input type="text" name="reg_name"
id="reg_name" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
```

```

        <label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Адреса</label>
        <input type="text" name="reg_adress"
id="reg_adress" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
        <label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Телефон</label>
        <input type="text" name="reg_phone"
id="reg_phone" class="col-12 col-sm-6 mx-1 col-md-5 col-lg-4">
        <label for="" class="col-12 col-sm-5 d-
flex justify-content-sm-end mb-0 pl-1 pb-sm-2">Посилання на фото</label>

        <input type="file" name="reg_photo"
multiple accept="image/*,image/jpeg" id="reg_photo" class="col-12 col-sm-6 mx-1 col-md-5 col-
lg-4">

        <div class="mt-2 col-6 offset-3 row
justify-content-center p-0">
                <button type="submit"
class="btn btn-danger btn-lg" name="reg_submit" id="form_submit">Регістрація</button>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</main>
<?php include('footer.php') ?>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
UIDAWAznBHeqEIIIVSCgzq+c9gqGAJn5c/t99JyeKa9xxaYpSvHU5awsuZVVFIhvj"
crossorigin="anonymous"></script>
<?php include('script.php') ?>
</body>
</html>

```

Додаток Г (index.php)

```

<?php
require_once 'database.php';
require_once 'function.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

```

```

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootst
rap.min.css"          rel="stylesheet"          integrity="sha384-
KyZXEAg3QhqLMPg8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We"
crossorigin="anonymous">
    <?php include('css.php') ?>
    <title>CLOCKS</title>
</head>
<body>
    <?php include('header.php') ?>
    <main>
        <div class="row m-0">
            <div class="col-12 col-md-4 p-0">
                <?php include('aside.php') ?>
            </div>
            <div class="main_section col-12 col-md-8">
                <?php include('block-parameter.php') ?>
                <div class="col-12 d-flex flex-column align-
items-center">
                    <h1>Популярне</h1>
                </div>
                <div class="row product_list">
                    <?php
                        $categories =
get_categories($link,"popular");
                    ?>
                    <?php foreach ($categories as $popular): ?>
                        <div class="col-lg-4 col-6 p-2" data-
sort="<?php echo $popular['price']?>">
                            <div class="wrapper col-12 p-2 d-flex
flex-column align-items-center justify-content-between">
                                <img <?=$popular["fotoUrl"] ?>
alt="comp" class="img-fluid">
                                    <h4><?=$popular["name"] ?></h4>
                                    <p><?=$popular["description"]
?></p>
                                <div class="price my-2">Ціна:
<span><?=$popular["price"] ?></span></div>
                                <div class="d-flex">
                                    <form method="POST"
action="add_to_wish.php" class="d-flex">
                                        <input type="text"
name="id" value='<?=$popular["id"] ?>' class="d-none">
                                        <input type="text"
name="category" value='popular' class="d-none">
                                        <button name="heart"
class="btn btn-primary mx-1">До обраного</button>
                                    </form>
                                    <form method="POST"
action="add_to_cart.php" class="d-flex">

```

```

        <input type="text"
name="id" value='<?=$popular["id"] ?>' class="d-none">
        <input type="text"
name="category" value='popular' class="d-none">
        <button type="submit"
name="cart" class="btn btn-success mx-1">До кошику</button>
    </form>
</div>

    </div>
</div>
<?php endforeach; ?>
</div>

</div>
</div>

</main>

<?php include('footer.php') ?>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstra
p.bundle.min.js" integrity="sha384-
U1DAWaznBHeqEIlVSCgzq+c9gqGAJn5c/t99JyeKa9xxaYpSvHU5awsuZVVFIhvj"
crossorigin="anonymous"></script>
<?php include('script.php') ?>
</body>
</html>

```

Додаток Е (script.js)

```

(function (factory) {
    if (typeof define === 'function' && define.amd) {
        // AMD (Register as an anonymous module)
        define(['jquery'], factory);
    } else if (typeof exports === 'object') {
        // Node/CommonJS
        module.exports = factory(require('jquery'));
    } else {
        // Browser globals
        factory(jQuery);
    }
})(function ($) {

    var pluses = /\+/g;

    function encode(s) {
        return config.raw ? s : encodeURIComponent(s);
    }

```

```

    }

    function decode(s) {
        return config.raw ? s : decodeURIComponent(s);
    }

    function stringifyCookieValue(value) {
        return encode(config.json ? JSON.stringify(value) :
String(value));
    }

    function parseCookieValue(s) {
        if (s.indexOf('"') === 0) {
            // This is a quoted cookie as according to RFC2068,
unescape...
            s = s.slice(1, -1).replace(/\\\"/g,
''').replace(/\\\\/g, '\\');
        }

        try {
            // Replace server-side written pluses with spaces.
            // If we can't decode the cookie, ignore it, it's
unusable.
            // If we can't parse the cookie, ignore it, it's
unusable.
            s = decodeURIComponent(s.replace(pluses, ' '));
            return config.json ? JSON.parse(s) : s;
        } catch(e) {}
    }

    function read(s, converter) {
        var value = config.raw ? s : parseCookieValue(s);
        return $.isFunction(converter) ? converter(value) : value;
    }

    var config = $.cookie = function (key, value, options) {

        // Write

        if (arguments.length > 1 && !$.isFunction(value)) {
            options = $.extend({}, config.defaults, options);

            if (typeof options.expires === 'number') {
                var days = options.expires, t = options.expires
= new Date();
                t.setMilliseconds(t.getMilliseconds() + days *
864e+5);
            }

            return (document.cookie = [
                encode(key), '=', stringifyCookieValue(value),

```



```

        options.expires ? '; expires=' +
options.expires.toUTCString() : '', // use expires attribute, max-
age is not supported by IE
        options.path ? '; path=' + options.path : '',
        options.domain ? '; domain=' + options.domain :
'',
        options.secure ? '; secure' : ''
    ].join(''));
    }

    // Read

    var result = key ? undefined : {},
        // To prevent the for loop in the first place assign
an empty array
        // in case there are no cookies at all. Also prevents
odd result when
        // calling $.cookie().
cookies = document.cookie ? document.cookie.split(';
') : [],
        i = 0,
        l = cookies.length;

    for (; i < l; i++) {
        var parts = cookies[i].split('='),
            name = decode(parts.shift()),
            cookie = parts.join('=');

        if (key === name) {
            // If second argument (value) is a function it's
a converter...
            result = read(cookie, value);
            break;
        }

        // Prevent storing a cookie that we couldn't decode.
        if (!key && (cookie = read(cookie)) !== undefined) {
            result[name] = cookie;
        }
    }

    return result;
};

config.defaults = {};

$.removeCookie = function (key, options) {
    // Must not alter options, thus extending a fresh object...
    $.cookie(key, '', $.extend({}, options, { expires: -1 }));
    return !$.cookie(key);
};

```

```

    ));

function cookie(){
    var name = $("#login").val();
    var description = $("#pass").val();
    $.cookie('login',name);
    $.cookie('pass',description);
}

function createCookie(name, value, days) {
    var expires;
    if (days) {
        var date = new Date();
        date.setTime(date.getTime() + (days * 24 * 60 * 60 *
1000));
        expires = "; expires=" + date.toGMTString();
    }
    else {
        expires = "";
    }
    document.cookie = name + "=" + value + expires + "; path=/";
}

function getCookie(c_name) {
    if (document.cookie.length > 0) {
        c_start = document.cookie.indexOf(c_name + "=");
        if (c_start != -1) {
            c_start = c_start + c_name.length + 1;
            c_end = document.cookie.indexOf(";", c_start);
            if (c_end == -1) {
                c_end = document.cookie.length;
            }
            return unescape(document.cookie.substring(c_start,
c_end));
        }
    }
    return "";
}

function exituser(){
    createCookie('user','',-1);
}
function mySortUp(){
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {
            if (+nav.children[i].getAttribute('data-
sort')>+nav.children[j].getAttribute('data-sort')){
                replaceNode = nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }
}

```

```

        }
    }
}
function mySortDown() {
    var nav=document.querySelector('.product_list');
    for (var i = 0; i < nav.children.length-1; ++i) {
        for (var j = i+1; j < nav.children.length; ++j) {

            if (+nav.children[i].getAttribute('data-
sort')<+nav.children[j].getAttribute('data-sort')){
                replaceNode = nav.replaceChild(nav.children[j],
nav.children[i]);
                insertAfter(replaceNode, nav.children[i]);
            }
        }
    }

    function insertAfter(elem, refElem){
        return
refElem.parentNode.insertBefore(elem,refElem.nextSibling);
    }

    var show=0;
    function block_show() {

        var ss=document.getElementById('block-sort');
        if (show==0)
        {
            show=1;
            ss.style.display='grid';
        }else{
            show=0;
            ss.style.display='none';
        }
    }
    function mySortPrice() {
        var nav=document.querySelector('.product_list');
        var
min=document.querySelector('#minprice').getAttribute('value');
        var
max=document.querySelector('#maxprice').getAttribute('value');

        for (var i = 0; i < nav.children.length; ++i) {
            if (+nav.children[i].getAttribute('data-sort')<min ||
+nav.children[i].getAttribute('data-sort')>max) {
                nav.children[i].style.display='none';
            }
            else
                nav.children[i].style.display='flex';
        }
    }
}

```

```

}
function inputMaxVal(a){
  $('#maxprice').attr('value',a.value);
}
function inputMinVal(a){
  $('#minprice').attr('value',a.value);
}

```

Додаток Ж(style.css)

```

.main_section{
  min-height: 100vh;
}
header{
  background: #333;
  background: linear-gradient(0deg, rgba(1,18,10,1) 0%,
  rgba(9,121,88,1) 37%, rgba(0,212,255,1) 100%);
  background-size: 200% 200%;

  -webkit-animation: Animation 5s ease infinite;
  -moz-animation: Animation 5s ease infinite;
  animation: Animation 5s ease infinite;
}
@-webkit-keyframes Animation {
  0%{background-position:10% 0%}
  50%{background-position:91% 100%}
  100%{background-position:10% 0%}
}

@-moz-keyframes Animation {
  0%{background-position:10% 0%}
  50%{background-position:91% 100%}
  100%{background-position:10% 0%}
}

@keyframes Animation {
  0%{background-position:10% 0%}
  50%{background-position:91% 100%}
  100%{background-position:10% 0%}
}
li{
  list-style-type: none;
}
h1{
  color: #fff;
}
.aside{
  z-index: 100;
  position: sticky;
  top:110px;
  background-image:linear-gradient(rgba(255, 255, 255,
0.8),rgba(255,151, 151, 0.5) );
  max-width: 325px;
}

```

```

border: 1px solid black;
max-height: 80vh;
border-radius: 0 25px 25px 0;
}
@media(max-width: 768px){
  .aside{
    max-width: 100vh;
  }
}
.aside a{
text-decoration: none;
font-weight: 600;
font-size: 24px;
color: #333;
}
.aside li{
border: 1px solid black;
}
.scrollmenu {
  overflow: auto;
  white-space: nowrap;
}
.scrollmenu::-webkit-scrollbar {
  height: 0px;
  width: 0;
}
.scrollmenu {
  -ms-overflow-style: none;
}

.scrollmenu {
  overflow: -moz-scrollbars-none;
}
.wrapper{
border: 2px solid black;
border-radius: 5px;
box-shadow: 0 0 10px black;
height: 100%;
background-color: #fff;
}
.wrapper:hover{
background-color: #aaa;
transition-duration:1s;
}

.price{
font-weight: 600;
}

.wrapper img{
height: 200px;
}

```

```

.wrapper p{
  overflow: hidden;
  text-overflow: ellipsis;
  display: -webkit-box;
  -webkit-line-clamp: 3;
  -webkit-box-orient: vertical;
}
footer{
  background-color: #b7b7b7;

  background: linear-gradient(0deg, rgba(0,212,255,1) 0%,
  rgba(9,121,88,1) 67%, rgba(1,18,10,1) 100%);
  background-size: 200% 200%;

  -webkit-animation: Animation 5s ease infinite;
  -moz-animation: Animation 5s ease infinite;
  animation: Animation 5s ease infinite;
}
.price{
  font-weight: 600;
}
img[alt="comp"]{
  height: 200px;
}
.hp a{
  color: yellow !important;
  text-decoration: none;
}
.navbar{
  background: transparent!important;
}

#wat1{
  top: -30px;
  left: 10px;
}
#Profile{
  top: -40px;
  left: 80px;
}
#wat2{
  bottom: -30px;
  right: 10px;
}
.slider{
  background: #FFFFFF;
  box-shadow: 0px 0px 50px rgba(64, 74, 84, 0.06);
  border-radius: 20px;
}
.circle{
  border: 0px!important;
}

```

```

border-radius: 7px;
width: 14px!important;
height: 14px!important;
background: #E0E0E0!important;
}
.circle.active{
background: #CFB88C!important;
}
#arrow_left2{
display: none;
}
.carousel-control-prev:hover #arrow_left{
display: none;
}
.carousel-control-prev:hover #arrow_left2{
display: flex;
}
#arrow_right2{
display: none;
}
.carousel-control-next:hover #arrow_right{
display: none;
}
.carousel-control-next:hover #arrow_right2{
display: flex;
}
main{
background-image:url(../img/bg.jpg);
background-repeat: no-repeat;
background-size: cover;
min-height: 100vh;
}

.menu a{
text-decoration: none;
color: orange;
font-size: 24px;
}
.wrapper2, .wrapper3{
border: 2px solid black;
border-radius: 5px;
box-shadow: 0 0 10px black;

background-color: #fff;
}
.wrapper2 img{
height: 100px;
}
#block-sort{
display: none;
}

```