

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

_____ червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «**Інформаційне та програмне забезпечення інтернет-магазину
відеоігор «GamesWorld»»**

здобувача групи ІН - 91 Бублика Владислава Олеговича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Владислав БУБЛИК
(підпис)

Керівник, доцент,

кандидат фізико-математичних наук

Сергій ШАПОВАЛОВ

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-91 Бублика Владислава Олеговича

1. Тема роботи: «Інформаційне та програмне забезпечення інтернет-магазину відеоігор»
затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань
2) Огляд методів розробки 3) Програмна реалізація 4) Аналіз результатів.
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 « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	Аналіз проблеми предметної області, постановка й формування завдань		
2	Огляд методів розробки		
3	Програмна реалізація		
4	Аналіз отриманих результатів		
5	Оформлення пояснювальної записки до кваліфікаційної роботи		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 88 стор., 64 рис., 4 додатки, 15 використаних джерел.

Обґрунтування актуальності теми роботи – веб-сайти інтернет магазинів є зручним інструментом для ведення приватного бізнесу. Велика кількість користувачів інтернету та зручність такої форми покупки товарів формують їх високу популярність. Конкуренція в сфері інтернет-магазинів вимагає розуміння методів створення та управління веб-сайтами.

Об'єкт дослідження — веб-сайт інтернет магазину

Мета роботи — інформаційна і програмна реалізація веб-сайту інтернет-магазину

Методи дослідження — технології створення веб-додатка.

Результати — розроблено веб-сайт для продажу популярних відеоігор на ігрові консолі, який має усі необхідні функції для його користування. Використовувались такі мови програмування як: HTML, CSS, JavaScript. Для формування списку товарів був використаний JSON. Для відправки форм дзвінка й замовлень на електронну пошту – API від FormSpree.

ВЕБ-САЙТ, ІНТЕРНЕТ МАГАЗИН, CSS, FRONT-END, HTML,
JAVASCRIPT, JSON

ЗМІСТ

ВСТУП	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД	7
1.1.Сучасний стан.....	7
1.2.Аналіз принципів побудови веб-сайтів інтернет-магазинів.....	8
1.3.Постановка задачі.....	14
2 ВИБІР МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ	16
2.1.Вибір стеку технологій.....	16
2.2.Визначення структури інтернет-магазину.....	22
2.3.Діаграма прецедентів.....	24
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	27
3.1 Підготовка до проекту.....	27
3.2 Програмна реалізація.....	31
ВИСНОВКИ	55
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	56
ДОДАТОК А	58
ДОДАТОК Б	68
ДОДАТОК В	81
ДОДАТОК Г	87

ВСТУП

Актуальність: у сучасному світі стрімкий розвиток технологій і висока популярність використання всесвітньої мережі відкривають величезний простір для створення й користування веб-сайтами. Актуальність роботи обумовлена необхідністю розуміння процесу їх розробки та вивчення підходів до їх реалізації.

Об'єкт дослідження: актуальність веб-сайтів інтернет-магазинів та їх вплив на бізнес-середовище. Веб-сайти зараз є одним з найзручніших способів для створення приватного бізнесу. Це можуть бути корпоративні веб-сайти, блоги, портали, соціальні мережі та інтернет-магазини. Останні є однією з найпопулярніших форм бізнесу в інтернеті. Це платформи, на яких відбувається продаж товарів або послуг у всесвітній мережі. Вони є зручними як для клієнтів, оскільки дозволяють їм здійснювати покупки в будь-який час та в будь-якому місці з доступом в мережу, так і для продавців, оскільки виключають більшість виплат за оренду приміщень, персонал, ін. [10]

Предмет дослідження: методи створення веб-сайтів та їх основні елементи, такі як: веб-дизайн, функціональність та взаємодія з користувачем. Інтернет-магазини зазвичай мають різні функціональні можливості, такі як каталог, пошук товарів за категоріями, фільтрація за параметрами, відгуки покупців, кошик та інші. Успіх інтернет-магазину залежить від багатьох факторів. Потрібно мати зрозумілий та привабливий інтерфейс, який забезпечує зручну взаємодію, високу швидкодію, успішну маркетингову стратегію та надання якісного обслуговування. Створення веб-сайту - це складний процес, що вимагає ретельного планування, проектування, розроблення та управління. Існує багато методів створення веб-сайтів. Основна їх різниця полягає в способі розробки та широті можливостей. Вибір способу залежить від потреб та цілей проекту. Основними методами є конструктори веб-сайтів та веб-розробка за допомогою HTML, CSS та JavaScript. Перший не вимагає знань з написання коду й для реалізації потреб

використовує шаблони, але через це має обмежені можливості. Другий – потребує програмування. HTML використовується для верстки веб-сайту, CSS - для його стилізації, а JavaScript – для роботи функцій.

Гіпотеза: на основі досліджень можна припустити, що ефективно використання сучасних методів розробки веб-сайтів сприятиме покращенню їх функціональності та взаємодії з користувачем.

Новизна: дослідження сучасних тенденцій та методів у сфері веб-розробки для покращення функціональності, дизайну та взаємодії з користувачем.

Структура: дана робота складається зі вступу, аналітичного огляду, постановки задачі, вибору методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Сучасний стан

Веб-сайт - це колекція веб-сторінок, які доступні через мережу Інтернет. Він може складатися з різних елементів, таких як текст, зображення, відео, аудіо, гіперпосилання та інші медіа-ресурси. Ці сторінки зв'язані між собою і зазвичай містять інформацію про певну тему, послуги, продукти або будь-яку іншу інформацію. Веб-сайти можуть мати різні цілі та функціональність. Вони використовуються для особистих цілей, таких як блоги або портфоліо, або для комерційних цілей, таких як інтернет-магазини, корпоративні сайти, новинні портали, соціальні мережі тощо.[11]

Інтернет-магазини є невід'ємною частиною сучасного електронного бізнесу та мають значний вплив на комерційну сферу життя. Це електронна платформа, яка надає можливість здійснювати продаж товарів або послуг споживачеві через мережу інтернет. Він дозволяє покупцям переглядати асортимент товарів, здійснювати покупки, а також забезпечує опції оплати та доставки. Веб-сайти забезпечують зручність та доступність для споживачів, а також надають багато переваг для торговців. Основними привілеями для покупців є можливість переглядати характеристики та відгуки товарів, порівнювати ціни та здійснювати покупки в будь-якому місці з доступом в інтернет та у будь-який час. Продавцям веб-сайти дозволяють розширити аудиторію, ефективно управляти товарами та моніторити замовлення. Відсутність необхідності у фізичних магазинах дозволяє виключити витрати на оренду приміщень та персонал.

Інтернет-магазини класифікуються за різними критеріями:

1. Метод роздрібного продажу: інтернет-магазини, веб-вітрини, проекти
2. За бізнес-моделлю: онлайн-магазин або онлайн-магазин суміщений з офлайн бізнесом
3. За взаємодією з постачальником: магазини з власним складом або магазини, які співпрацюють з постачальниками

4. За відсотком автоматизації: веб-вітрини, інтернет-магазини та торгові Інтернет-системи (TIS). [5]

Сучасні тенденції розробки веб-сайтів полягають у впровадженні нових технологій та їх розвитку для поліпшення користувацького досвіду та візуального стилю. Зараз велика увага приділяється адаптивному дизайну для коректного відображення на всіх пристроях, красивим анімаціям та інтерактивності. Логічним продовженням є використання штучного інтелекту для покращення взаємодії користувачів із сайтом.

1.2. Аналіз принципів побудови веб-сайтів інтернет-магазинів

Інтернет магазини мають стандартизований вигляд і структуру для зручності користування. Вони, зазвичай, включають в себе такі важливі елементи:

- Головна сторінка: це сторінка, на яку потрапляють відвідувачі, відкриваючи сайт. Вона повинна мати привабливий дизайн і надавати коротку інформацію про магазин, товари або послуги, спеціальні пропозиції, новини чи акції.
- Каталог товарів: це розділ зі списком товарів, в якому можна здійснювати пошук чи фільтрацію по певним параметрам.
- Сторінки товарів: кожний товар має свою власну сторінку з детальною інформацією про нього, включаючи зображення, опис, характеристики, ціну, наявність та відгуки користувачів.
- Кошик: це функціональний елемент, що дозволяє покупцям додавати товари до замовлення, визначати їх кількість та переглядати перед оформленням покупки.
- Форма оформлення замовлення: це сторінка, на якій користувачі вводять свої контактні дані, адресу доставки, спосіб оплати та іншу інформацію, необхідну для оформлення замовлення.

Створення веб-сайтів – поетапний процес, що, в більшості випадках, вимагає планування, проектування й розробки. Основні етапи створення веб-сайту, зазвичай, включають такі пункти:

1. Збір вимог: визначення мети вашого веб-сайту, цільової аудиторії, функціональних можливостей, дизайну та інших аспектів.
2. Планування: визначення структури, розміщення сторінок, навігації, функціональних елементів та інших деталей.
3. Дизайн: визначення візуального дизайну веб-сайту: вибір кольорової палітри, шрифтів, графічних елементів та інших компонентів.
4. Розробка: програмна реалізація веб-сайту: розробка фронт-енду (HTML/CSS/JavaScript) та бек-енду (серверна логіка, база даних, інтеграції тощо).
5. Тестування і відладка: тестування веб-сайту на наявність помилок, перевірка оптимізації, швидкодії, безпеки.
6. Реліз: процес запуску та введення в дію веб-сайту на живому сервері для забезпечення його доступності користувачам Інтернету.
7. Підтримка: виправлення помилок, оновлення функціональності, додавання нового контенту, розширення можливостей та інші модифікації після релізу. [5]

Підтримка є важливою частиною життєвого циклу веб-сайтів, особливо для бізнесів і компаній. Для полегшення управління ними була створена система управління контентом (CMS). Це система, яка дозволяє користувачам створювати, редагувати та впорядковувати контент на веб-сайті без необхідності програмування.

CMS надає зручний інтерфейс, який дозволяє власникам та адміністраторам веб-сайтів легко керувати різними аспектами свого сайту. Так можуть додаватися статичні сторінки, створюватися нові статті та дописи,

керуватися зображення та медіафайли, налаштовуватися навігаційні меню, а також керуватися користувачі та їхні права доступу.

Однією з головних переваг CMS є можливість швидко оновлювати та змінювати вміст веб-сайту в режимі реального часу. Крім того, CMS дозволяє використовувати шаблони та розширення для налаштування зовнішнього вигляду та функціональності веб-сайту. [8]

У сучасному конкурентному цифровому середовищі, веб-сайт потребує правильного просування для привернення уваги цільової аудиторії, а саме досягненні високого рівня відвідуваності, покращенні його позицій у пошукових системах та підвищенні рівня конверсії. Ключові елементи просування інтернет-магазину включають:

- Цифровий маркетинг: використання різних каналів цифрового маркетингу, таких як пошукова оптимізація (SEO) та оплата за клік (PPC).
- Соціальні медіа: присутність на популярних платформах соціальних мереж, таких як Facebook, Instagram і Twitter.
- Електронна пошта: надсилання персоналізованих пропозицій, новин та рекламних матеріалів підписникам.
- Інфлюенсер-маркетинг: співпраця з впливовими особистостями віртуального простору, які мають велику аудиторію і популярність в певній галузі. Це може включати спонсорські пости, огляд товарів, рецензії та рекомендації.
- Рекламні кампанії: використання різних форм онлайн-реклами, таких як банерна реклама, реклама на соціальних медіа, платна реклама в пошукових системах. [8]

Юзабіліті (користувальницька зручність) є важливим аспектом веб-сайтів, оскільки відноситься до властивостей, що визначають ефективність та

задоволення користувачів при використанні веб-сайту. Основна мета полягає в тому, щоб забезпечити максимально легкий доступ до інформації та функцій сайту, а також сприяти задоволенню користувачів під час їх взаємодії з сайтом.

Основні елементи юзабіліті включають:

- Простота використання: веб-сайт повинен мати інтуїтивно зрозумілу навігацію та логічну структуру.
- Ефективність: веб-сайт повинен дозволяти користувачам виконувати свої завдання швидко та без зайвих зусиль.
- Читабельність та зрозумілість: текст повинен бути легкочитаним та зрозумілим.
- Консистентність: веб-сайт повинен мати єдиний стиль та дизайн на всіх сторінках.
- Відповідність контексту: інформація та функції на веб-сайті повинні бути відповідні контексту та потребам користувачів.
- Відсутність помилок: веб-сайт повинен бути перевірений на наявність помилок і ті, в разі їх присутності, мають бути виправлені. [8]

Зв'язок в інтернет-магазинах грає важливу роль у взаємодії між продавцями і покупцями. За відсутності фізичного контакту, онлайн зв'язок стає особливо важливим для забезпечення задоволення клієнтів та вироблення позитивного враження про бренд.

Основні аспекти зв'язку в інтернет-магазинах включають:

- Зручність зв'язку: інтернет-магазини повинні забезпечити зручні та доступні засоби зв'язку з клієнтами. Це може включати електронну пошту, онлайн-чат, телефонну підтримку або соціальні медіа.
- Швидка відповідь: важливо, щоб інтернет-магазини реагували на запити клієнтів швидко і ефективно.

- Якісна підтримка: надання якісної підтримки клієнтам допомагає вирішувати їхні проблеми, відповідати на запитання та надавати необхідну інформацію.
- Система сповіщень: інтернет-магазини можуть використовувати електронні листи або повідомлення для забезпечення актуальної інформації клієнтам. Це може бути повідомлення про стан замовлення, спеціальні пропозиції або повідомлення про нові товари.
- Відгуки та оцінки: зворотній зв'язок від клієнтів у вигляді відгуків та оцінок може бути важливим елементом зв'язку в інтернет-магазинах.
- Соціальні медіа: інтернет-магазини можуть використовувати платформи соціальних медіа для спілкування з клієнтами, відповіді на запитання, надання інформації про акції та новини. [8]

Швидкодія веб-сайту є одним з ключових факторів для забезпечення успіху веб-проекту. Вона впливає на загальний досвід користувача, конверсію, рангування в пошукових системах та багато інших аспектів. Основна мета продуктивності полягає в тому, щоб сторінки завантажувалися якомога швидше і миттєво реагували на дії користувачів. Декілька ключових факторів, які впливають на швидкодію веб-сайту:

- Оптимізація зображень: зображення часто становлять значну частину завантаження сторінки. Важливо оптимізувати зображення за допомогою стиснення, вибору відповідного формату та розміру, використанням невисокої якості та кешуванням.
- Кешування: кешування дозволяє зберігати деякі ресурси, такі як зображення, стилі, скрипти, на більш доступному місці, зменшуючи час завантаження для повторних відвідувачів.
- Кешування браузера: кешування браузера дозволяє зберігати деякі ресурси на комп'ютері користувача, що дозволяє прискорити завантаження сторінок при повторному відвідуванні.

- Мініфікація файлів: мініфікація файлів, таких як HTML, CSS та JavaScript, полягає у видаленні зайвих пробілів, коментарів та зменшенні обсягу коду.
- Мінімізація запитів: зменшення кількості запитів, які веб-сайт виконує до сервера, може позитивно вплинути на швидкодію. Це можна досягти шляхом об'єднання файлів CSS та JavaScript, використання спрайтів для зображень та використання кешування для статичних ресурсів.
- Вибір швидкого хостингу: вибір надійного та швидкого хостинг-провайдера важливий для швидкодії веб-сайту. [8]

Візуальний аспект дизайну, його організація та взаємодія з користувачем мають великий вплив на успіх веб-сайту. Основні елементи дизайну веб-сайту включають:

- Інформаційна архітектура: ефективна організація і структура веб-сайту дозволяють користувачам легко зорієнтуватися і знайти потрібну інформацію.
- Візуальний дизайн: використання привабливого та згуртованого дизайну, який відповідає бренду і стилістиці компанії, є важливим елементом у користувацькому досвіді.
- Використання мультимедіа: включення відео, зображень, анімацій та інших мультимедійних елементів може допомогти зробити веб-сайт цікавим та привабливим для користувачів.
- Зручність взаємодії: дизайн веб-сайту повинен забезпечувати зручну взаємодію користувачів з сайтом.
- Консистентність стилю: усі сторінки веб-сайту повинні мати консистентний стиль та дизайн, що відповідають бренду компанії. [8]

Отже, дослідивши основні моделі інтернет-магазинів, важливо запровадити такі можливості як: перегляд інформації про магазин, каталог,

пошук товару, кошик, оформлення замовлення товару й зворотній зв'язок. Важливими елементами також будуть привабливий візуальний вигляд, зручність й простота взаємодії та швидкодія.

1.3. Постановка задачі

Метою проєкту є створення веб-сайту інтернет-магазину популярних відеоігор на ігрові консолі. Все, що стосується адміністрування, буде відповідати потребам продавця. А саме, товари будуть додаватися в каталог через масив об'єктів в json файлі, а замовлення будуть приходити у вигляді форм на електронну пошту. Магазин призначений для цільової аудиторії геймерів, які користуються ігровими консолями. Для них будуть впроваджені привабливий вигляд із зрозумілим інтерфейсом та достатня кількість функціональних можливостей для зручного користування.

Веб-сайт буде мати наступні такі основні елементи:

- Шапка: розміщена на всіх сторінках веб-сайту і має лого, посилання на сторінки та кнопку телефону.
- Головна сторінка: на ній розміщена основна інформація про магазин, яка включає загальний опис, відповіді на популярні питання, розділ з найпопулярнішими товарами, де можна переглянути їх трейлери, форма для замовлення дзвінку й контакті дані продавця.
- Каталог: на ній представлений список товарів в наявності, можливості фільтрації по символам і додавання в кошик. При чому, багаторазове натискання змінює кількість товару в кошику.
- Кошик: умовно поділений на дві частини екрану. Зліва має список доданих товарів, з можливістю змінювати їх кількість та видаляти, а справа форму для заповнення даних покупця, підраховану суму замовлення й кнопку підтвердження. При чому, якщо кошик пустий,

замовлення оформити неможливо, а зліва текст вказує на необхідність додати товар у каталозі.

Товари в каталозі будуть представлені у вигляді зображення обкладинки гри, назви, ціни й ціни зі знижкою, якщо така є. Ціна розміщена на кнопці, що додає його в каталог. В кошику до цих параметрів додається кількість одиниці товару. Детальна інформація, що містить жанр, розробника, серію є тільки на найпопулярніших товарах на головній сторінці.

Отже, розробка веб-сайту інтернет-магазину відеоігор з цільовою аудиторією геймерів має на меті надати зручний і привабливий інструмент для покупки популярних товарів для ігрових консолей, враховуючи потреби покупців, а також зручності адміністратора.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1. Вибір стеку технологій

При створенні веб-сайту інтернет-магазину, вибір методу розробки є важливим етапом, який суттєво вплине на підсумковий результат. Так як основною метою є створення привабливого та зручного інтерфейсу й впровадження великої кількості функцій, метод має бути гнучким.

Існує багато інструментів для розробки веб-сайтів, такі як: онлайн-платформи, контент-менеджери, фреймворки, або веб-розробка за допомогою мов html, css, javascript, ін.

- **Онлайн-платформи:** мають вбудовані функціональні можливості для створення веб-сайтів. Зазвичай, розробка відбувається за допомогою шаблонів, які заповнюються під свої потреби. Спосіб є легким, так як не потребує програмування, але дуже обмеженим. Веб-сайт може не виділятися серед конкурентів та не виконувати певних потреб. Також, недоліком є те, що онлайн-платформи дозволяють користуватися їх повними версіями лише за гроші. Відомими представниками є: WIX, Tilda, Jimdo.

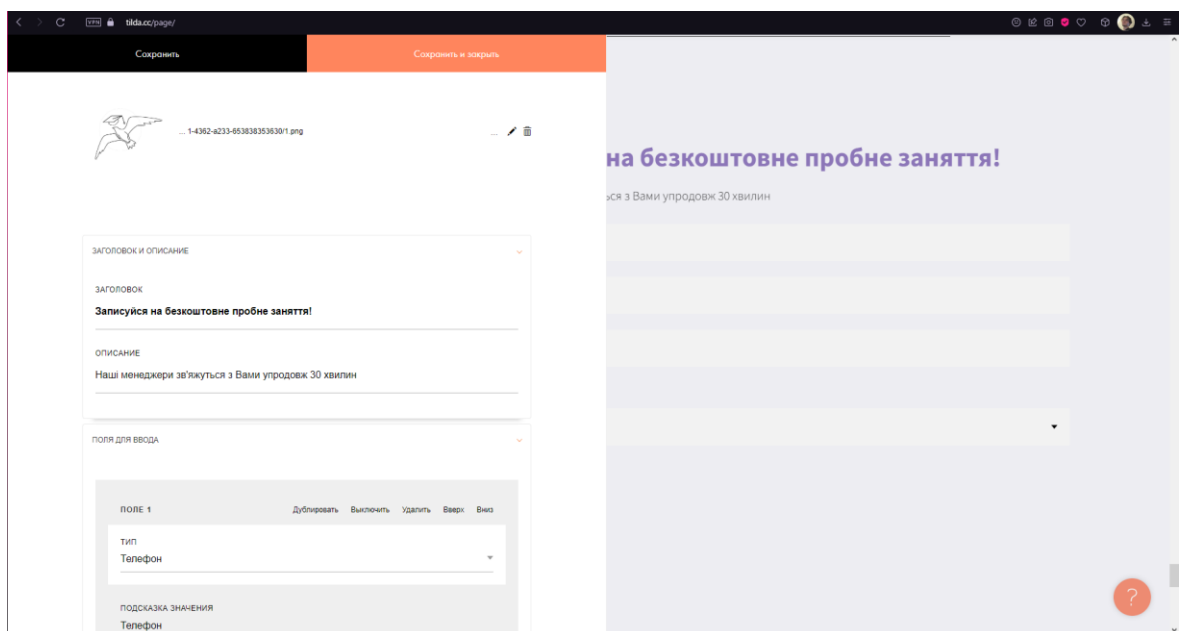


Рисунок 2.1 – Інтерфейс платформи Tilda

- **Контент-менеджери:** дозволяють створювати веб-сторінки й управляти ними за допомогою шаблонів та розширень. Вони мають зручний інтерфейс для додавання контенту та редагування сторінок. Є складнішими за онлайн-платформи, але й мають більшу функціональність. Відомими представниками є: WordPress, Joomla, Drupal.

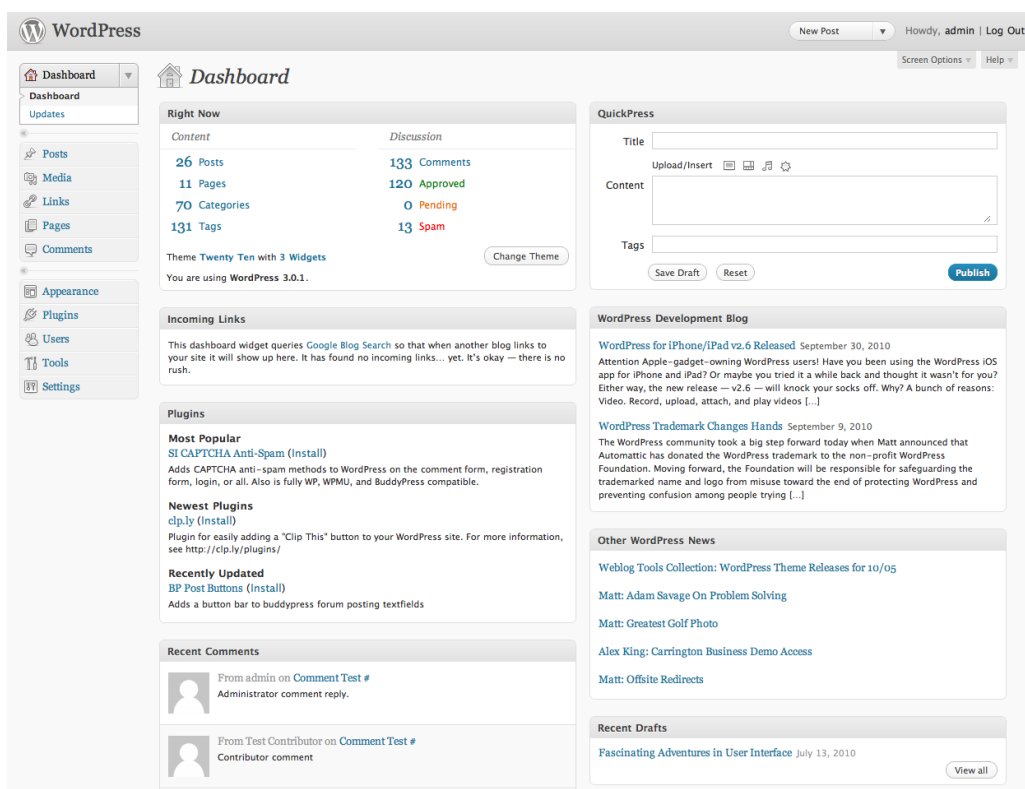


Рисунок 2.2 – Інтерфейс контент-менеджера WordPress

- **Фреймворки:** надають розширені можливості для розробки веб-сайтів. Вони забезпечують набір інструментів, що допомагають впоратися зі складними завданнями та допомагають зменшити кількість коду. Через це потребують глибоких знань з веб-розробки. Відомими представниками є: Angular, React, jQuery.

- Веб-розробка: передбачає розробку з нуля без використання шаблонів та фреймворків. Використовуються такі мови, як: HTML, CSS, JavaScript, PHP, Python та ін. Цей метод надає найбільшу гнучкість та контроль над розробкою, але вимагає знань з програмування й більшого часу.

Для реалізації веб-сайту інтернет магазину була обрана саме веб-розробка з нуля і такі інструменти як: HTML, CSS, JavaScript та JSON. Цей стек технологій надає бажану гнучкість для створення привабливого зручного інтерфейсу, а також впровадження потрібних функцій для роботи каталогу, кошика, кнопок, форм й інших елементів.

- HTML (HyperText Markup Language) – мова розмітки веб-сторінок. Використовується для структурування й організації вмісту веб-сторінки. Елементи на сторінці позначаються як заголовки, абзаци, зображення, форми, кнопки за допомогою тегів. [1]

```
<!DOCTYPE html>
<html>
<body>

<h1>Мій перший заголовок</h1>
<p>Мій перший параграф.</p>

</body>
</html>
```

Рисунок 2.3 – Приклад HTML коду

- CSS (Cascading Style Sheets) – відповідає за стилізацію та зовнішній вигляд веб-сторінок. В ньому визначаються розташування елементів, кольори, шрифти, зображення та їх зовнішній вигляд. Стилi задаються за допомогою певних параметрів і властивостей і накладаються на теги, що були зазначені в HTML. [2]

```

body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}

```

Рисунок 2.4 – Приклад CSS коду

- JavaScript – є мовою програмування, яка відповідає за функції на веб-сторінках. Використовується для створення логічних функцій з обробкою даних, асинхронного завантаження, роботи з API, JSON, формами, кнопками, створення анімацій та багато іншого. Також допомагає взаємодіяти з сервером. [3]

```

<!DOCTYPE html>
<html>
<body>

<h2>Мій перший JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Натисніть мене, щоб відобразити дату та час.</button>

<p id="demo"></p>

</body>
</html>

```

Рисунок 2.5 – Приклад JavaScript коду

- JSON (JavaScript Object Notation) – є форматом обміну даними, який використовується для збереження та передачі структурованих даних. Використовується для взаємодії між веб-сервером і веб-клієнтом і для збереження даних в файловій системі. Представляє дані у вигляді пар "ключ-значення". [13]

```
{  
  "employees": [  
    {"firstName": "John", "lastName": "Doe"},  
    {"firstName": "Anna", "lastName": "Smith"},  
    {"firstName": "Peter", "lastName": "Jones"}  
  ]  
}
```

Рисунок 2.6 – Приклад JSON коду

Тож, веб-сайт інтернет-магазину відеоігор буде реалізовано тільки з використанням front-end. Це частина веб-додатку, яка взаємодіє з користувачем і відповідає за відображення і інтерактивність контенту.

Back-end, що є серверною частиною і відповідає за збереження інформації, обробку даних, взаємодією з базою даних, в даному випадку не потребується. Це обумовлено тим, що інтернет-магазин спеціалізується на продажі тільки найпопулярніших відеоігор на ігрові консолі, які активно обговорюються в інформаційному просторі. Це означає, що кількість товарів є невеликою і база даних для них не потребується. Також, вони не містять значної кількості детальної інформації, оскільки самі диски не відрізняються фізично, а про, власне, ігри цільова аудиторія не потребує більшості даних, оскільки вже з ними ознайомена. Використання лише front-end спрощує розробку й забезпечує зручну платформу з достатньою кількістю можливостей без зайвих рішень.

За відправку даних форм замовлення дзвінку й товарів на електронну пошту продавця відповідає API від FormSpree.

API (Application Programming Interface) - це набір правил та протоколів, які дозволяють взаємодіяти між різними програмами або сервісами.

FormSpree є платформою, яка надає можливість для обробки форм на веб-сайтах і відправки їх вмісту на вказану електронну пошту. Після заповнення форми і натисненні кнопки відправки, дані з форми передаються

на сервери FormSpree, а потім відправляються як повідомлення на вказану електронну пошту.

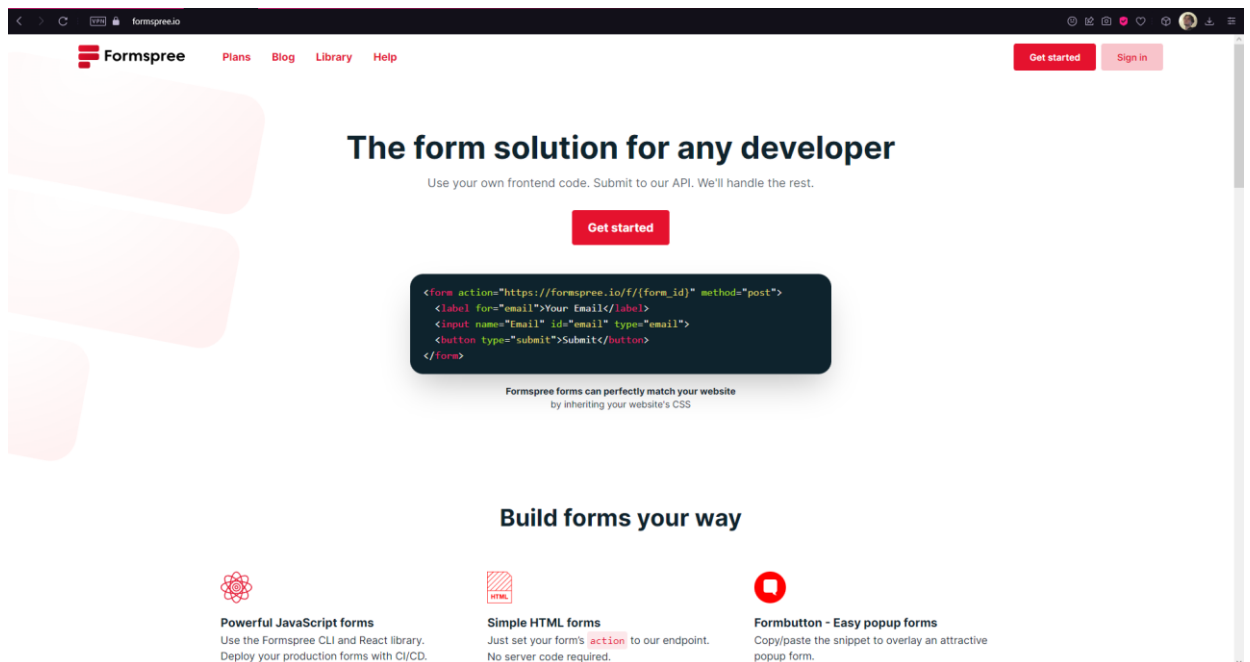


Рисунок 2.7 – головна сторінка платформи FormSpree

Як середовище розробки веб-сайту використовувалась програма RHPStorm від компанії JetBrains. Це інструмент розробки веб-додатків, які можна створити за допомогою PHP, HTML, JavaScript і CSS.

Перевагами RHPStorm є висока продуктивність та широкий спектр функцій, які спрощують процес розробки. Він надає зручний інтерфейс, автодоповнення коду, розуміння структури проекту. Також дана середовище розробки має багато інструментів для рефакторингу, налагодження й тестування.

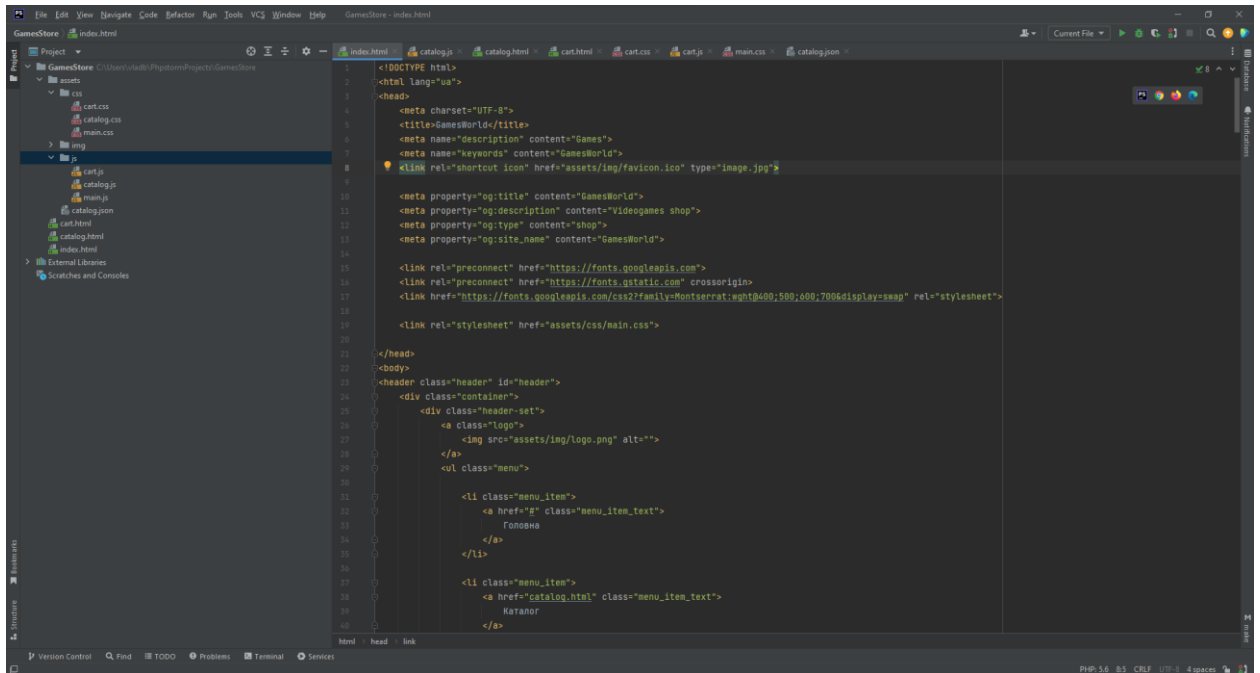


Рисунок 2.8 – Інтерфейс середі розробки PhpStorm

Отже, під час розробки, HTML використовується для розмітки головної сторінки, каталогу й кошику. CSS – для створення бажаного дизайну елементів на цих сторінках. JavaScript – для впровадження функцій для роботи кнопок, форм, завантаження каталогу, кошику, додавання, видалення й зміни кількості товарів, підрахування суми замовлення та відправки даних форм на електронну пошту продавця за допомогою API від FormSpree. JSON використовується для формування списку товарів у вигляді масиву.

2.2. Визначення структури інтернет-магазину

При розробці інтернет-магазину відеоігор «GamesWorld», важливим етапом є визначення структури, яка забезпечує логічний та зручний інтерфейс для користувачів. Структура магазину допомагає організувати навігаційні елементи, функціональність і взаємопов'язати сторінки.

Існують різні типи внутрішніх структур, які можна використовувати в інтернет-магазинах:

- Лінійна структура: сторінки магазину послідовно посилаються одна на одну, утворюючи лінійний шлях для користувача. Цей підхід є простим та зрозумілим для користувачів, добре підходить для невеликих магазинів без великої кількості сторінок.

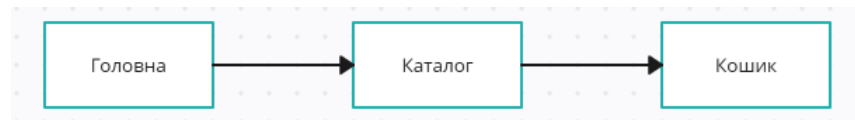


Рисунок 2.9 – Приклад лінійної структури

- Ієрархічна структура: кожна сторінка може мати підсторінки. Цей підхід, наприклад, дозволяє групувати товари за категоріями.

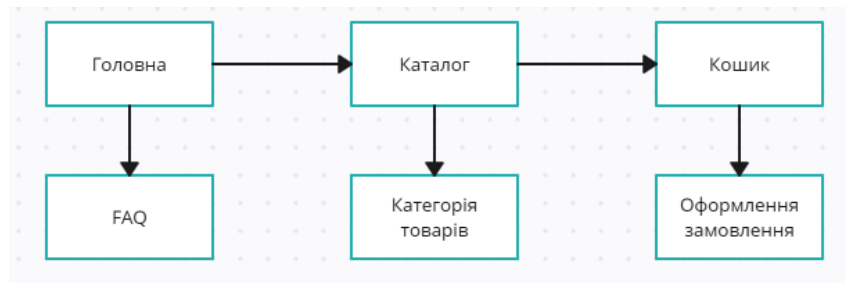


Рисунок 2.10 – Приклад ієрархічної структури

- Мережева структура: сторінки магазину зв'язані між собою не тільки лінійно або ієрархічно, але й утворюють мережу з багатьма зв'язками. Наприклад, з головної сторінки можна перейти в каталог, з каталогу – до категорії товарів, а звідти - до акційних пропозицій.



Рисунок 2.11 – Приклад мережевої структури

- Комбінована структура: поєднує різні типи структур. Наприклад, можна застосовувати лінійну структуру на головній сторінці і ієрархічну структуру в каталозі.

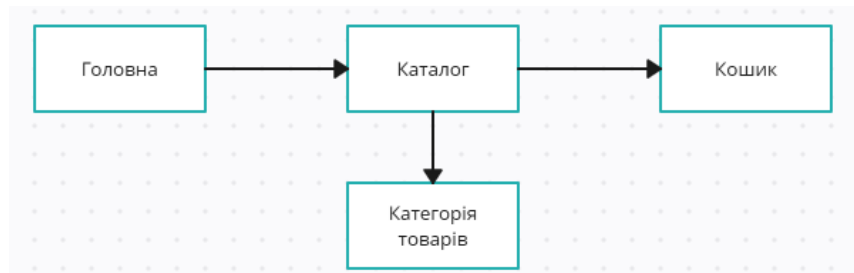


Рисунок 2.12 – Приклад комбінованої структури

Для веб-сайту інтернет магазину «GamesWorld» була обрана лінійна структура, де сторінки послідовно посилаються одна на одну. Головна сторінка містить посилання на каталог, каталог вказує на необхідність переходу в кошик під час додавання товару, з кошику відбувається оформлення замовлення. Пустий кошик посилається на необхідність переходу у каталог. Окрім того, на кожній сторінці є шапка з посиланнями на всі сторінки, що також забезпечує легку навігацію для користувачів.



Рисунок 2.13 – Структура магазину «GamesWorld»

2.3. Діаграма прецедентів

Діаграми прецедентів є одним з типів діаграм стандарту UML (Unified Modeling Language). Він використовується для моделювання програмного забезпечення і дозволяє візуалізувати процеси і роботи систем. Основна мета діаграм прецедентів - визначити, які дії можуть бути виконані акторами системи та які результати вони приносять. Це допомагає встановити вимоги до системи і визначити взаємодію користувачів з додатком.

Діаграми прецедентів складаються з акторів, прецедентів та взаємодій між ними.

- Актори – це зовнішні сутності, які взаємодіють з системою, наприклад, користувачі, зовнішні системи або інші програми.
- Прецеденти – це окремі функціональні можливості системи, які можуть бути виконані акторами.
- Взаємодія – показує, як актори взаємодіють з прецедентами, описує послідовність дій між ними. [9]

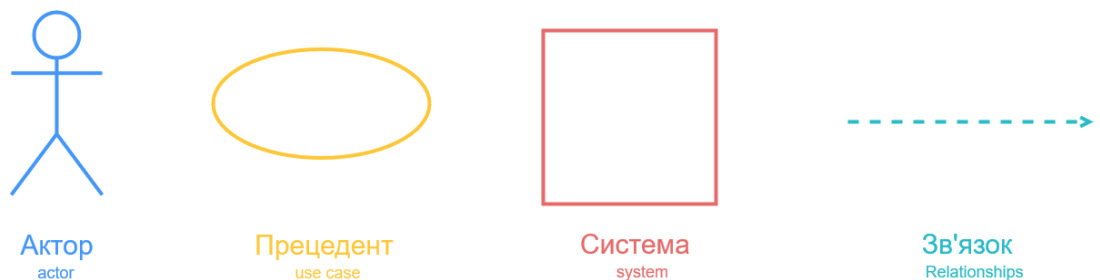


Рисунок 2.14 – Основні позначення діаграми варіантів використання

Діаграми взаємодії в UML вказують на порядок виконання дій між об'єктами взаємодії.

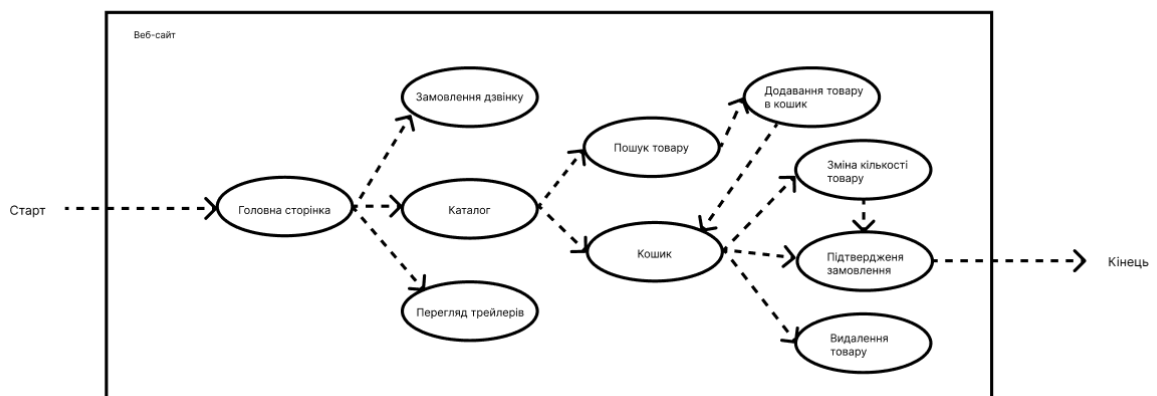


Рисунок 2.15 – Діаграма взаємодії

Діаграми використання в UML використовуються для візуалізації функціональних можливостей системи і способів їх використання акторами.

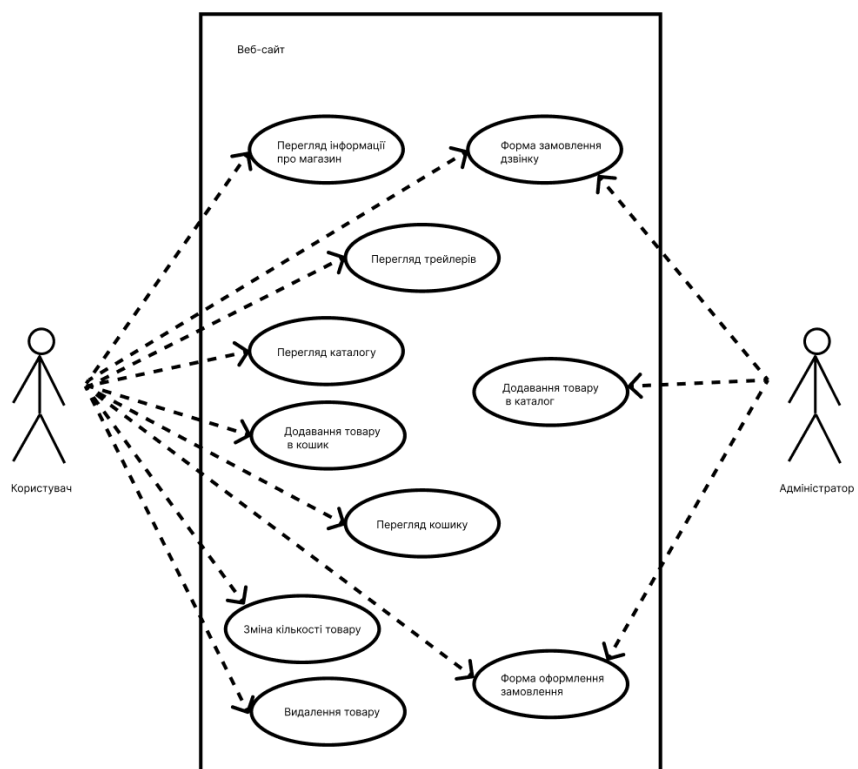


Рисунок 2.16 – Діаграма використання

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Підготовка до проекту

Першим кроком у підготовці до розробки є встановлення середовища розробки PhpStorm від компанії JetBrains. Для цього було завантажено встановлювальний файл з офіційного веб-сайту JetBrains і виконано процес установки згідно з інструкціями.

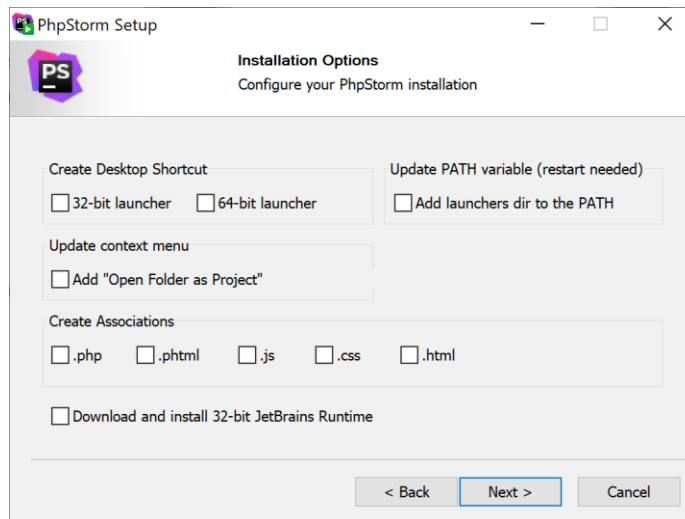


Рисунок 3.1 – Вікно встановлення PhpStorm

Далі, важливим етапом підготовки було визначення дизайну, основних кольорів і шрифтів для веб-сайту. Зовнішній вигляд повинен бути привабливим, зручним для користувачів і відповідати цілям проекту. Також важливо, щоб він був гармонійним для своєї тематики.

Основна ідея дизайну сторінок веб-сайту інтернет магазину «GamesWorld» полягає у їх поділенні на уявні тематичні блоки, які містять мінімум тексту й велику кількість зображень, які будуть інтуїтивно описувати деталі. На рисунку 3.2, 3.3, 3.4 зображено макети сторінок веб-сайту, які показує цей принцип. Сірі форми є уявними блоками, зелені форми відповідають за зображення, жовті - за форму.

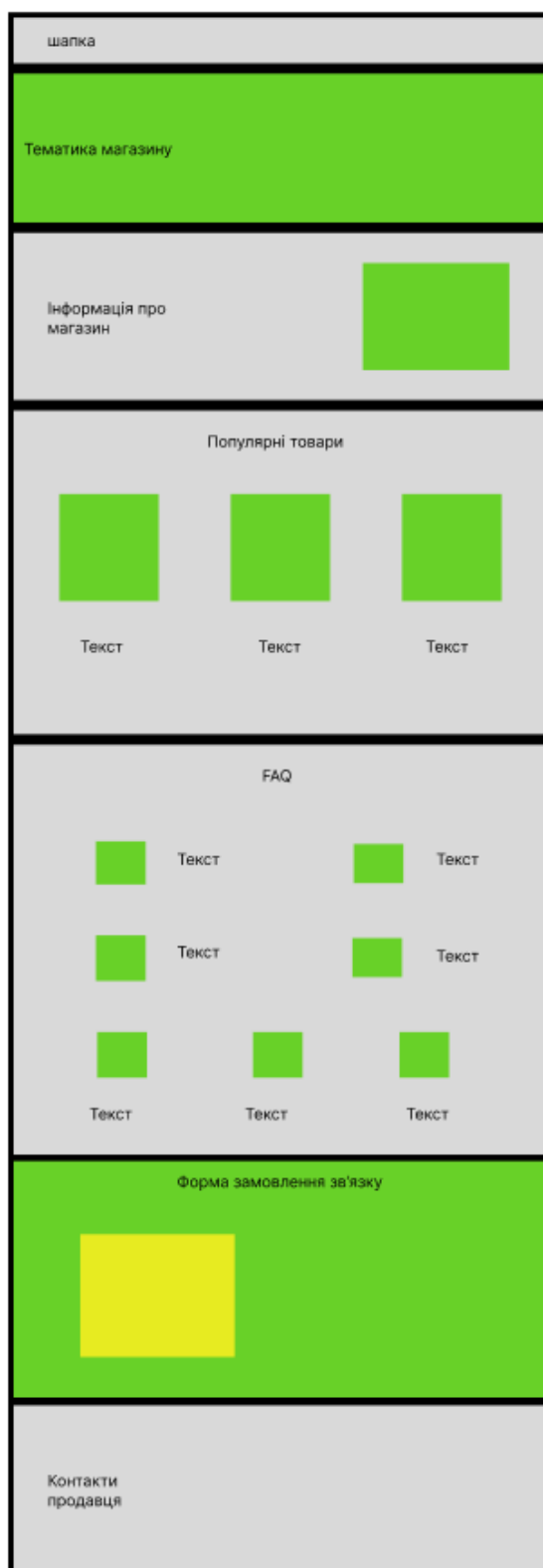


Рисунок 3.2 – Макет головної сторінки веб-сайту

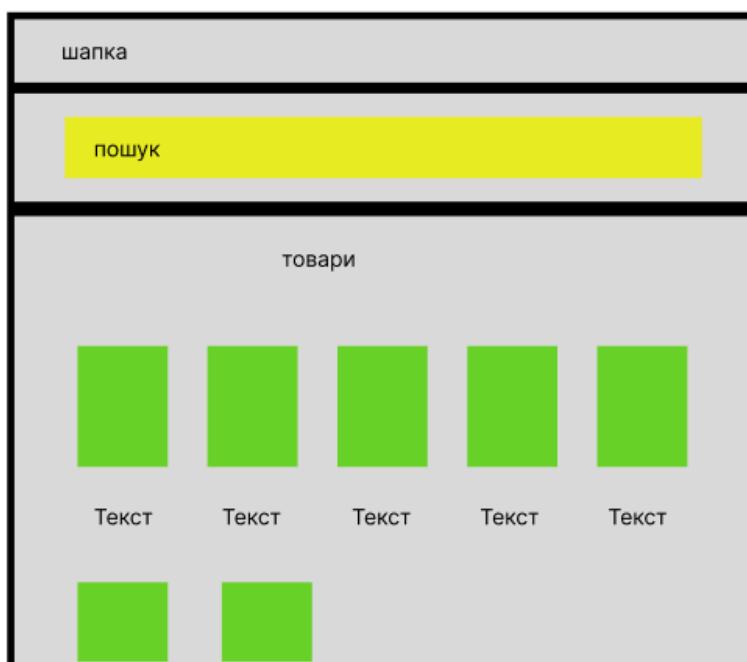


Рисунок 3.3 – Макет сторінки каталогу веб-сайту

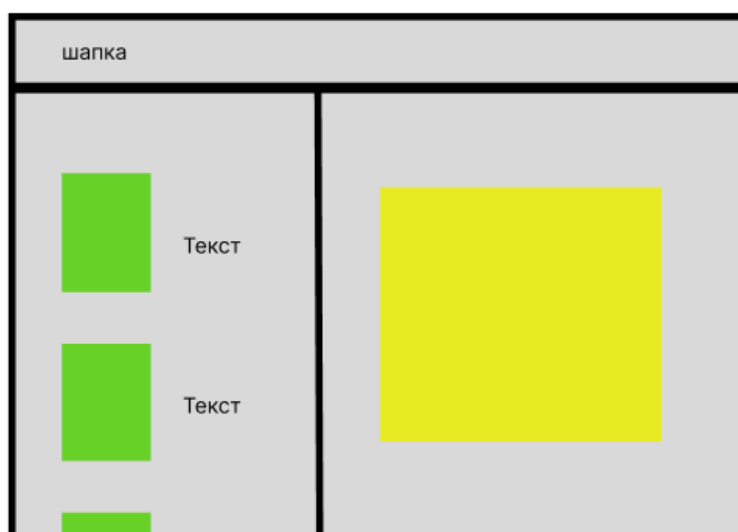


Рисунок 3.4 – Макет сторінки кошику веб-сайту

Зображення використовуються для логотипу, іконок, фонових зображень, а також представлення товарів.

Основною кольоровою гамою були обрані зелені й жовті кольори, а саме:



Рисунок 3.5 – Кольорова палітра веб-сайту

Були обрані сучасні, читабельні шрифти, а саме сімейство шрифтів **Montserrat**.

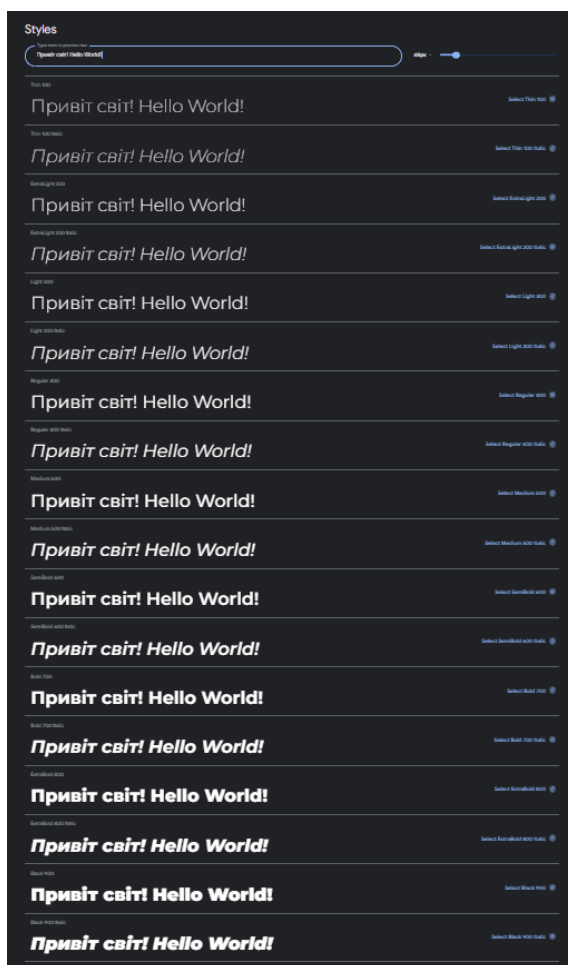


Рисунок 3.6 – Приклад зовнішнього вигляду шрифту **Montserrat**

Сукупність цих рішень поєднані тематикою веб-сайта і створюють дружню гармонійну атмосферу.

3.2 Програмна реалізація

Після запуску PhpStorm створюється папка проекту з наступною ієрархією файлів. Файли різних типів розміщуються в окремих директоріях для зручності подальшої роботи.

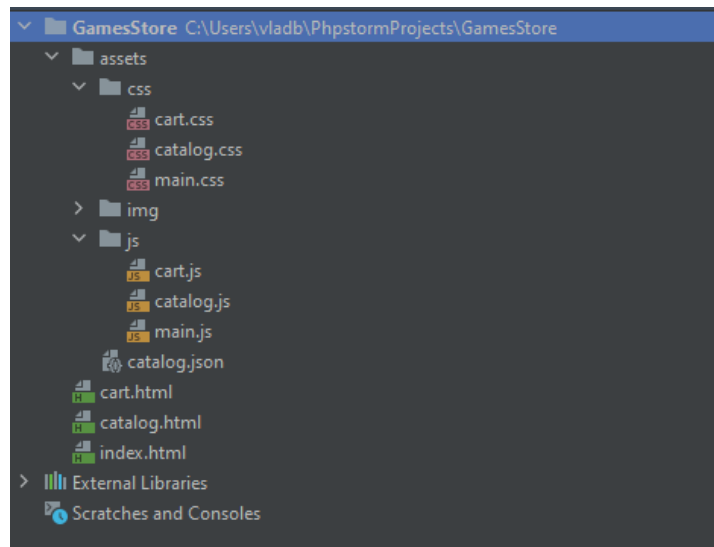


Рисунок 3.7 – Ієрархія файлів проекту

Попередньо завантажуються зображення, що знадобляться для реалізації проекту, в папку `img`.

Далі розпочинається написання коду за допомогою `html`, `css` і `javascript`.

Основним `html` файлом є `index.html`. Він відповідає за метадані, підключення шрифтів, стилів, функцій та розмітку головної сторінки. Приклад заповнення `head` зображено на рисунку 3.8

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>GamesWorld</title>
  <meta name="description" content="Games">
  <meta name="keywords" content="GamesWorld">
  <link rel="shortcut icon" href="assets/img/favicon.ico" type="image.jpg">

  <meta property="og:title" content="GamesWorld">
  <meta property="og:description" content="Videogames shop">
  <meta property="og:type" content="shop">
  <meta property="og:site_name" content="GamesWorld">

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;700&display=swap" rel="stylesheet">

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

```

Рисунок 3.8 – Код заповнення head в index.html

Головна сторінка містить такі блоки:

- Шапка з посиланнями на сторінки: головна, каталог, кошик й номер телефону

```

<header class="header" id="header">
  <div class="container">
    <div class="header-set">
      <a class="logo">
        
      </a>
      <ul class="menu">
        <li class="menu_item">
          <a href="#" class="menu_item_text">
            Головна
          </a>
        </li>
        <li class="menu_item">
          <a href="catalog.html" class="menu_item_text">
            Каталог
          </a>
        </li>
        <li class="menu_item">
          <a href="cart.html" class="menu_item_text">
            Кошик
          </a>
        </li>
      </ul>
      <a href="tel:+380666037907" class="phone">
        066-603-7907
      </a>
    </div>
  </div>
</header>

```

Рисунок 3.9 – Код шапки в index.html

- Тематика магазину з кнопкою переходу в каталог

```

<section class="info" id="info">
  <div class="container">
    <div class="info-set">
      <h1 class="title">
        Магазин комп'ютерних ігор
      </h1>
      <h3 class="info-title">
        Відкрийте для себе нові <span class="info-span">світи!</span>
      </h3>
      <p class="info-text">
        Найнижчі ціни
      </p>
      <a href="catalog.html" class="btn">
        Каталог
      </a>
    </div>
  </div>
</section>

```

Рисунок 3.10 – Код «тематики магазину» в index.html

- Про нас

```

<section class="about" id="about">
  <div class="container">
    <div class="about-set">
      <div class="about-text">
        <div class="about_wrap">
          Про нас
        </div>
      </div>
      <h2 class="about_title">
        Ігри - це наше <span class="text-span-2">натхнення</span>
      </h2>
      <p class="about_text">
        Кожен знаходить в відеоіграх щось рідне для себе. Для когось це лише розвага після важкого дня, для когось - окремі світи. А для нас - це ще робота, яку ми з радістю виконуємо, щоб подарувати вам незабутні враження! Ми є невеликим приватним магазином, що тільки починає свій шлях, але сподіваємося, що принесемо вам лише позитивні емоції. Телефонуйте та робіть замовлення, щоб скоріше поринути у неймовірний віртуальний світ!
      </p>
    </div>
    
  </div>
</section>

```

Рисунок 3.11 – Код «Про нас» в index.html

- Популярні товари з кнопками перегляду трейлерів та кнопкою переходу в каталог

```

</section>
<section class="popular" id="popular">
  <div class="container">
    <h2 class="title-2">
      Популярні товари
    </h2>
    <p class="text-2">
      Зараз у тренді
    </p>
    <div class="item-set">
      <div class="item">
        
        <div class="item-column">
          <h4 class="item_title">
            Call Of Duty: Modern Warfare 2
          </h4>
          <p class="item_text">
            > Видавець: Activision
          </p>
          <p class="item_text">
            > Розробник: Infinity Ward, Raven Software
          </p>
          <p class="item_text">
            > Серія: Call of duty
          </p>
          <p class="item_text">
            > Жанр гри: Шутер від першої особи (FPS)
          </p>
          <a href="https://www.youtube.com/watch?v=r72GP1PIZa0" class="item_button">Трейлер</a>
        </div>
      </div>
    </div>
  </div>
</section>

```

Рисунок 3.12 – Код одиниці популярного товару в index.html

- FAQ з якорем на форму замовлення дзвінка

```

<section class="faq" id="faq">
  <div class="container">
    <h2 class="title-2">
      FAQ
    </h2>
    <p class="text-2">
      Магазин GamesWorld знаходиться в м.Суми , але ми працюємо - <span
      class="text-span">по всій Україні.</span>
    </p>
    <div class="faq-set">
      <div class="set">
        <div class="pic">
          
          <div class="faq-kit">
            <h4 class="faq_title">
              Час доставки
            </h4>
            <p class="faq_text">
              Відправка відбувається у день замовлення, якщо воно було зроблене до 17:00
            </p>
          </div>
        </div>
        <div class="pic">
          
          <div class="faq-kit">
            <h4 class="faq_title">
              Упаковка
            </h4>
            <p class="faq_text">
              Всі замовлення запаковуються в пузирчатку плівку, що забезпечує цілісність товару
            </p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Рисунок 3.13 – Код одного набору питань в «FAQ» в index.html

- Форма замовлення дзвінка. Відправка форми реалізована за допомогою API від FormSpree.

```

<section class="call" id="call">
  <div class="container">
    <div class="order-set">
      <div class="form-wrap">
        <h4 class="form-title">
          Замовити дзвінок
        </h4>
        <p class="form-text">
          Заповніть поля знизу, і ми зв'яжемося з вами, щоб обговорити всі питання
        </p>
        <form action="https://formspree.io/f/mayzkond" method="POST">
          <label>
            <input type="text" placeholder="Ім'я*" name="Ім'я" class="input input-name" id="nameId"
              required>
            <input type="email" placeholder="Email" name="email" class="input">
            <input type="tel" placeholder="Телефон*" name="Телефон" class="input" id="phoneId" required>
            <textarea placeholder="Додатковий коментар" name="Коментар" class="input input-text"></textarea>
            <button type="submit" class="btn-form" id="btn-form">Надіслати</button>
          </label>
        </form>
      </div>
      <div class="ellipse">
        <div class="ellipse-1"></div>
        <div class="ellipse-2"></div>
        <div class="ellipse-3"></div>
        <div class="call-text">
          <div class="call_text_wrap">
            Приємної гри
          </div>
          <p class="call_text_wrap-2">
            Ми цінуємо кожного покупця та з нетерпінням чекаємо на ваше повернення
          </p>
        </div>
      </div>
    </div>
  </div>
</section>

```

Рисунок 3.14 – Код форми замовлення дзвінка в index.html

- Контактні дані продавця

```

<footer class="footer" id="footer">
  <div class="container">
    <h2 class="title-3">
      Інформація про нас
    </h2>
    <p class="text-3">
      Наші контакти
    </p>
    <div class="contacts">
      <p class="contact-text">
        Бублик Владислав Олегович
      </p>
      <div class="contact-set">
        <a href="mailto:vladpublic@gmail.com" class="contact-email">vladpublic@gmail.com</a>
        <a href="tel:0666037907" class="contact-phone">+380666037907</a>
      </div>
      <p class="contact-text">
        Україна М.Суми
      </p>
    </div>
    <div class="social">
      <a href="https://www.facebook.com/profile.php?id=100007657141376" class="social-item">
        
      </a>
      <a href="https://www.instagram.com/vladpublic/" class="social-item">
        
      </a>
      <a href="https://twitter.com/VBublic" class="social-item">
        
      </a>
      <a href="https://www.linkedin.com/in/vladyslav-bublyk-3247951b9/" class="social-item">
        
      </a>
    </div>
    <p class="text-4">
      ©GamesWorld 2023
    </p>
  </div>
</footer>

```

Рисунок 3.15 – Код контактної інформації в index.html

Далі для дизайну й правильного розміщення всіх елементів на головній сторінці створюється файл `main.css`. До тегів застосовуються певні параметри й опції. На рисунку 3.16 показано приклад задання параметрів для блоків з тематикою магазину й «Про нас», які містять налаштування розташування об'єктів, тексту, кольорів, кнопок.

```
.info-title {
  font-weight: 600;
  font-size: 33px;
  line-height: 40px;

  color: #FFFFFF;
}

.info-span {
  color: #4fd529;
}

.info-text {
  color: rgba(255, 255, 255, 0.9);
  padding: 30px 0;
}

.btn {
  display: block;
  background: #4fd529;
  border-radius: 30px;
  color: #ffffff;
  padding: 17px 30px;
  width: 130px;
}

.btn:hover {
  color: #000000;
  background: #00ff59;
}

.about {
  padding: 125px 0;
}

.about-set {
  display: flex;
  align-items: center;
  justify-content: space-between;
}
```

Рисунок 3.16 –Код розділів тематики й «Про нас» в `main.css`

Потім створюється файл `main.js`. В ньому реалізується функція для роботи кнопки у формі замовлення дзвінка. Вона оброблює надсилання форми за допомогою AJAX, надсилаючи дані на сервер без перезавантаження сторінки. Після успішного надсилання форми, форма

скидається, і виводиться повідомлення про успіх. У випадку помилки під час надсилання виводиться повідомлення про помилку.

```
const form = document.querySelector( selectors: 'form');
const submitButton = document.querySelector( selectors: '#btn-form');

form.addEventListener( type: 'submit', listener: function (event :SubmitEvent ) {
    event.preventDefault();

    const formData = new FormData(form);

    const xhr = new XMLHttpRequest();
    xhr.open( method: 'POST', form.action, async: true);
    xhr.setRequestHeader( name: 'Accept', value: 'application/json');
    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                form.reset();
                alert('Чекайте на дзвінок!');
            } else {
                alert('Вибачте. Сталася помилка!');
            }
        }
    };
    xhr.send(formData);
});
```

Рисунок 3.17 – Функція роботи кнопки в main.js

В результаті виходить головна сторінка, що задовольняє всі потреби і має такий зовнішній вигляд:

GAMSWORLD Головна Каталог Кошик 066 403 7907

Магазин комп'ютерних ігор

Відкрийте для себе нові світи!

Дізнайтеся більше

Ігри – це наше НАТХНЕННЯ

Компи знаходять в магазині щось відрізняє себе. Для когось це лише довгий після важкого дня, для когось – це місце, де можна відпочити, а для нас – це те, що робить нашу роботу цікавішою. Ми не вважали за необхідне, щоб посприяти вам у вашому відпочинку. Ми є незалежним приватним магазином, що тільки починає свій шлях, але сподіваємося, що прокинемо вам новий погляд на ігри. Телефонуйте та робіть замовлення, щоб скористатися першими унікальними пропозиціями!

Популярні товари

Зображення ігор:

- Call of Duty: Modern Warfare 2**
 - Видавець: Activision
 - Розробник: Infinity Ward, Raven Software
 - Серія: Call of Duty
 - Жанр: Тримісний шутер від першої особи (FPS)
- Horizon Forbidden West**
 - Видавець: Sony Interactive Entertainment
 - Розробник: Guerrilla Games
 - Серія: Horizon
 - Жанр: Різноманітний (RPG)
- Elden Ring**
 - Видавець: Bandai Namco Entertainment
 - Розробник: From Software
 - Серія: Dark Souls
 - Жанр: Різноманітний (RPG)

FAQ

Магазин GameWorld знаходиться в м.Суми, але ми працюємо – по всій Україні.

- Час доставки**
Відправка відбувається у день замовлення, якщо не вказано інакше, робота до 17:00.
- Упаковка**
Всі замовлення запаковані в надійну та надійну упаковку, щоб забезпечити цілісність товару.
- Оригінальність товару**
Всі товари є ліцензованими та пройшли перевірку.
- Оформлення замовлення**
Для оформлення замовлення вам достатньо заповнити форму або зателефонувати нам.

Доставка по всій Україні
Нова Платформа (Укр. Пошта), Мехі

Гарантія
Всі товари перевіряються протягом 2-х днів

Високий рейтинг від покупців
Тільки позитивні відгуки

Замовити дзвінок
Заповніть поле знизу, ми зв'яжемося з вами, щоб обговорити всі питання.

МІСЬ: _____
ЕМАЙЛ: _____
Телефон: _____
Додатковий коментар: _____

Ми цінуємо кожного покупця та з неперервним чеканням на ваше повернення.

Інформація про нас

Наші контакти

Булвар Вільності, Суми
viber: @gameworld.com.ua
066 403 7907

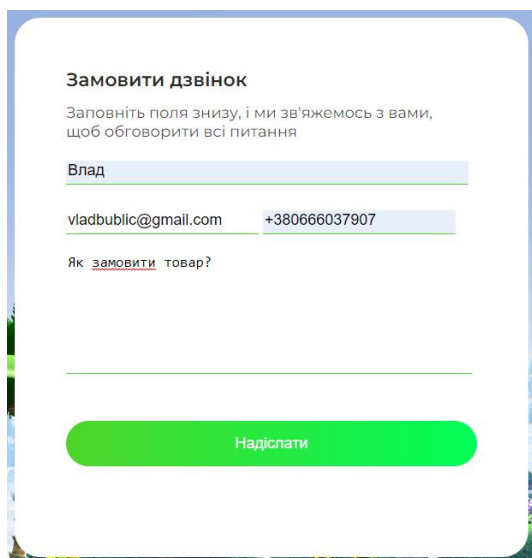
Україна м.Суми

f @ t in

©GameWorld 2023

Рисунок 3.18 – Зовнішній вигляд головної сторінки

Успішна робота форми замовлення дзвінка представлена на рисунках 3.19, 3.20, 3.21



Замовити дзвінок

Заповніть поля знизу, і ми зв'яжемося з вами, щоб обговорити всі питання

Влад

vladbublic@gmail.com +380666037907

Як **ЗАМОВИТИ** товар?

Надіслати

Рисунок 3.19 – Заповнена форма замовлення дзвінка

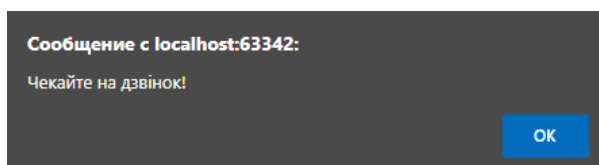


Рисунок 3.20 – Повідомлення при успішній роботі форми

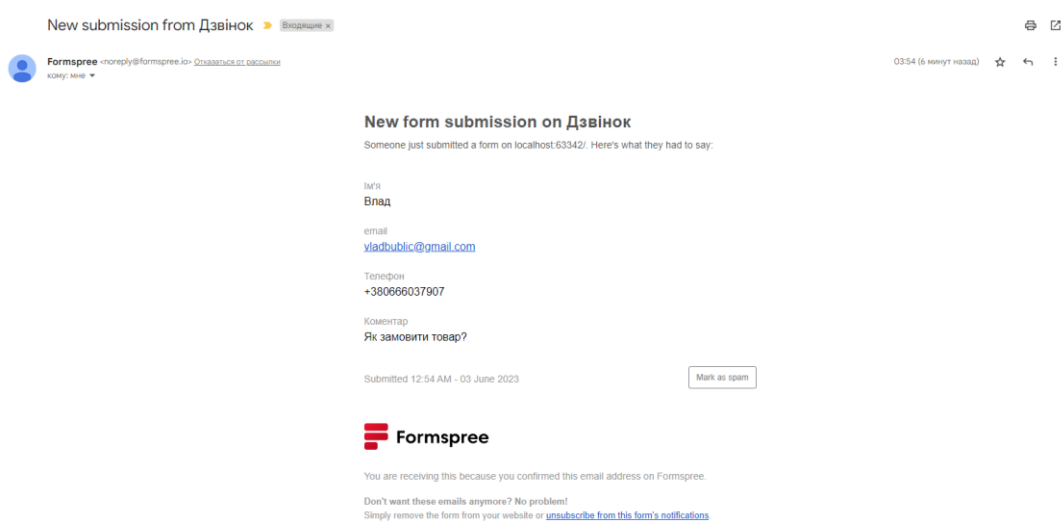


Рисунок 3.21 – Отримання даних форми замовлення дзвінка на електронну пошту

Другим етапом є створення сторінки каталогу.

Створюється файл `catalog.html`. Він містить підключення шрифтів, стилів та розмітку сторінки каталогу, що має такі блоки:

- Шапка
- Рядок пошуку

```
<section class="search">

  <label for="searchInput"></label>
  <input type="text" placeholder="Введіть назву товару" name="text" class="input input-search" id="searchInput">
  <button onclick="search()" class="btn-search"></button>

</section>
```

Рисунок 3.22 – Код рядку пошуку в `catalog.html`

- Список товарів

```
<section class="catalog" id="catalog">
  <div class="items">
  </div>
</section>
```

Рисунок 3.23 – Код списку товарів в `catalog.html`

Формування списку товарів відбувається за допомогою функції. Створюється json файл `catalog.json` з масивом об'єктів, з якими потім відбувається робота в javascript.

```
[
  {
    "image": "assets/img/img_callofduty.png",
    "name": "Call of Duty: Modern Warfare 2",
    "id": 1,
    "price": "2199"
  },
  {
    "image": "assets/img/img_horizon.png",
    "name": "Horizon: Forbidden West",
    "id": 2,
    "price": "2799",
    "newPrice": "1999"
  },
  {
    "image": "assets/img/img_eldenring.png",
    "name": "Elden Ring",
    "id": 3,
    "price": "1499"
  },
  {
    "image": "assets/img/thelastofus.jpg",
    "name": "The Last Of Us",
    "id": 4,
    "price": "2399"
  },
]
```

Рисунок 3.24 – Частина списку товарів в `catalog.json`

Створюється файл `catalog.js` для створення функцій завантаження даних з json й формування списку товарів, а також функцій роботи рядку пошуку і додавання товарів у кошик.

Функція завантаження даних з json `loadItemsFromJSON` працює таким чином, що спочатку встановлюється URL-адреса JSON-файлу, який містить дані каталогу товарів. Потім створюється об'єкт `XMLHttpRequest`, який взаємодіє з сервером і отримує дані. Запускається запит до сервера методом `send()` і відбувається відправка GET-запиту до JSON-файлу. При успішному отриманні даних у відповіді сервера, викликається функція `loadCatalog(catalog)`, яка буде формувати товари в каталозі.

```

window.addEventListener( type: 'load', loadItemsFromJSON);

let catalog;

1 usage
function loadItemsFromJSON() {
  const jsonFile = 'assets/catalog.json';

  const xhr = new XMLHttpRequest();
  xhr.overrideMimeType( mime: 'application/json');
  xhr.open( method: 'GET', jsonFile, async: true);
  xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      catalog = JSON.parse(xhr.responseText);
      loadCatalog(catalog);
    }
  };
  xhr.send();
}

```

Рисунок 3.25 –Завантаження даних з JSON в `catalog.js`

Функція `loadCatalog` знаходить контейнер, що буде мати елементи каталогу й далі формує товари. Кожна одиниця має зображення, назву, ціну й акційну ціну (якщо така є). Кнопка викликає функцію `addToCart`, що додає товар в кошик. Закінчується функція тим, що додає елементи в контейнер.

```

2 usages
function loadCatalog(items) {
  const itemsContainer = document.querySelector( selectors: '.items');
  itemsContainer.replaceChildren();
  items.forEach(function (item) {
    const itemElement = document.createElement( tagName: 'div');
    itemElement.classList.add('item');

    const imgElement = document.createElement( tagName: 'img');
    imgElement.src = item.image;
    imgElement.alt = '';
    imgElement.classList.add('item-img');
    itemElement.appendChild(imgElement);

    const titleElement = document.createElement( tagName: 'h4');
    titleElement.classList.add('item_title');
    titleElement.id = item.id;
    titleElement.textContent = item.name;
    itemElement.appendChild(titleElement);

    const buttonElement = document.createElement( tagName: 'a');
    buttonElement.href = item.orderLink;
    if (item.newPrice == null) {
      buttonElement.classList.add('item_button');
      buttonElement.textContent = item.price + ' грн';
    } else {
      buttonElement.classList.add('item-button-mod');
      const newPrice = document.createElement( tagName: 'span');
      newPrice.textContent = item.newPrice + ' грн';
      const price = document.createElement( tagName: 'span');
      price.classList.add('item-button-span-white');
      price.textContent = item.price + ' грн';

      buttonElement.appendChild(price);
      buttonElement.appendChild(newPrice);
    }

    buttonElement.addEventListener( type: 'click', listener: function (event :MouseEvent) {
      event.preventDefault();
      addToCart(item);
    });

    itemElement.appendChild(buttonElement);

    itemsContainer.appendChild(itemElement);
  });
}

```

Рисунок 3.26 –Формування каталогу товарів в catalog.js

Функція рядку пошуку search() отримує введені дані користувача, створює порожній масив, який буде містити знайдені товари і перевіряє, чи включає назва товару, перетворена у нижній регістр, символи з рядку пошуку, також перетворені у нижній регістр. При успішному знаходженні символів, товар додається до масиву searchResult. Якщо співпадінь немає – отримуємо повідомлення «Товар не знайдено».

```

1 usage
function search() {
  const search = document.getElementById('searchInput').value;
  let searchResult = [];
  catalog.forEach(function (item) {
    if (item.name.toLowerCase().includes(search.toLowerCase())) {
      searchResult.push(item);
    }
  });
  if (searchResult.length !== 0)
    loadCatalog(searchResult);
  else
    alert("Товар не знайдено");
}

```

Рисунок 3.27 – Функція рядку пошуку в catalog.js

Функція додавання товару в кошик addToCart() спочатку отримує, збережений у локальному сховищі, кошик товарів. Якщо він порожній, створюється порожній масив cart. Далі при додаванні товару, за ідентифікатором, шукається чи містить його кошик. Якщо товар існує, збільшується його кількість на одиницю, якщо ні – він додається з кількістю 1. Після натиснення кнопки виводиться «Товар додано до кошику!». Кошик зберігається у локальному сховищі.

```

function addToCart(item) {
  let cart = JSON.parse(localStorage.getItem('cart')) || [];
  console.log(cart);
  const existingItem = cart.find((cartItem) => cartItem.id === item.id);
  console.log(existingItem);
  if (existingItem) {
    existingItem.count += 1;
  } else {
    item.count = 1;
    cart.push(item);
  }
  console.log(cart);
  localStorage.setItem('cart', JSON.stringify(cart));
  alert("Товар додано до кошику!");
}

```

Рисунок 3.28 – Функція додавання товару в кошик в catalog.js

Потім, створюється файл `catalog.css` для встановлення дизайну й розташування всіх елементів. На рисунку 3.29 зображено частину коду, що відповідає за зовнішній вигляд списку товарів у каталозі.

```
.items {  
    display: flex;  
    flex-wrap: wrap;  
    margin-top: 50px;  
    flex-direction: row;  
    gap: 20px;  
}  
  
.item {  
    text-align: center;  
    width: 280px;  
    height: 430px;  
    position: relative;  
}  
  
.item .item-img {  
    margin-bottom: 0;  
    width: auto;  
    height: 230px;  
}  
  
.item .item_button, .item .item-button-mod{  
    position: absolute;  
    bottom: 0;  
    width: 200px;  
    justify-content: center;  
    align-items: center;  
}  
  
.catalog {  
    padding-top: 50px;  
    padding-bottom: 50px;  
    margin-left: 150px;  
    justify-content: center;  
    justify-items: center;  
    justify-self: center;  
}
```

Рисунок 3.29 – Код зовнішнього вигляду списку товарів в `catalog.css`

Таким чином, отримуємо сторінку каталогу, що задовольняє всім потребам дизайну й функціональності.

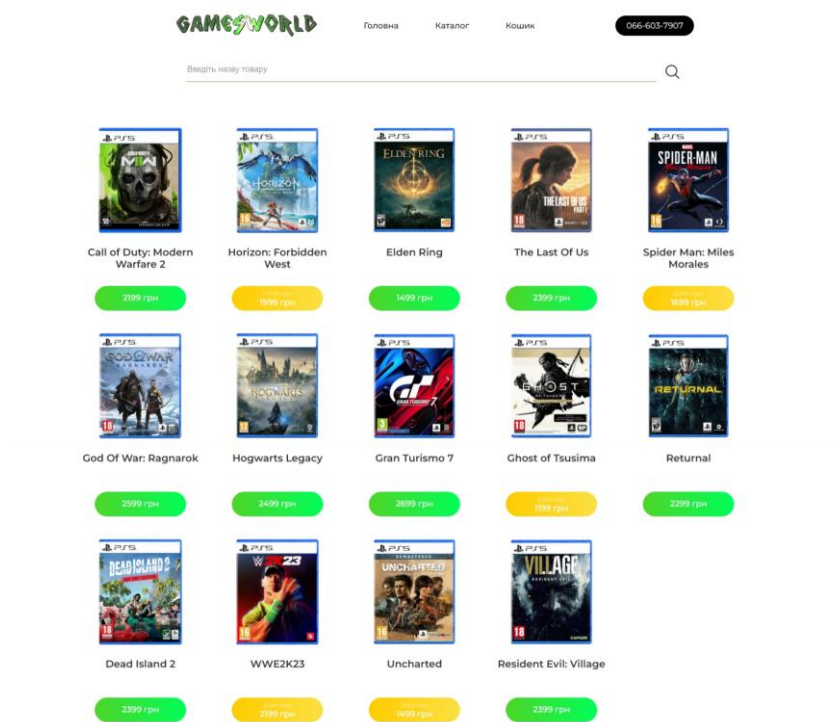


Рисунок 3.30 – Зовнішній вигляд сторінки каталогу

Успішна робота рядку пошуку представлена на рисунку 3.31



Рисунок 3.31 – Робота рядку пошуку

Успішна робота кнопок додавання товару в кошик представлена на рисунку 3.32

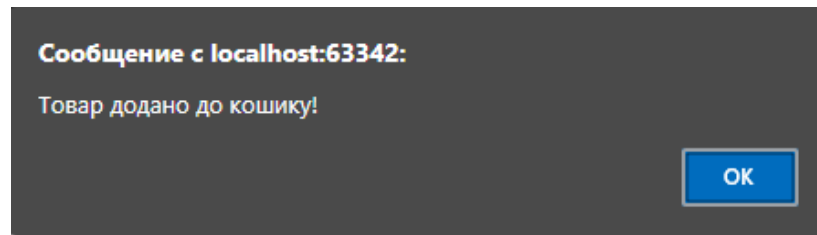


Рисунок 3.32 – Повідомлення про успішне додавання товару в кошик

Останньою сторінкою є кошик. Для нього створюється файл розмітки cart.html. Він також містить підключення шрифтів і стилів. Кошик містить такі блоки:

- Шапка
- Список доданих товарів

```
<div class="container">
  <div class="cart">
    <div class="items">
    </div>
  </div>
</div>
```

Рисунок 3.33 – Код доданих товарів в кошик в cart.html

- Форма оформлення замовлення з підрахованою ціною, дані з якої відправляються на електронну пошту за допомогою API FormSpree.

```
<div class="order">
  <form action="https://formspree.io/f/mknaerbw" method="POST" class="order-form">
    <h2 class="order-form-title">
      Оформлення замовлення
    </h2>
    <label>
      <input type="text" placeholder="Ім'я*" name="Ім'я" class="input" id="nameId" required>
      <input type="email" placeholder="Email" name="email" class="input">
      <input type="tel" placeholder="Телефон*" name="Телефон" class="input" id="phoneId" required>
      <input type="text" placeholder="Місто*" name="Місто" class="input" id="cityId" required>
      <input type="text" placeholder="Пошта*" name="Пошта" class="input" id="postId" required>
      <textarea placeholder="Додатковий коментар" name="Коментар" class="input input-text"></textarea>
    </label>
    <h2 class="total-price">
    </h2>
    <button type="submit" class="checkout-btn">Підтвердити замовлення</button>
  </form>
</div>
```

Рисунок 3.34 – Код форми замовлення в cart.html

Далі, для заповнення кошику товарами й впровадження функцій для його роботи створюється файл `cart.js`

Перш за все, створюється функція `loadCartFromJSON()`, що відповідає за отримання, збереженого у локальному сховищі, кошику товарів. Якщо він порожній, створюється пустий масив `cart`. Після отримання даних кошика викликається функція `loadCart()`, яка відповідає за відображення кошика на сторінці.

```

window.addEventListener( type: 'load', loadCartFromJSON);
window.addEventListener( type: 'load', updateTotal);

let cart;

1 usage
function loadCartFromJSON() {
    cart = JSON.parse(localStorage.getItem( key: 'cart')) || [];
    loadCart();
}

```

Рисунок 3.35 – Функція отримання даних кошику в `cart.js`

Далі створюється функція `loadCart()`. Вона знаходить контейнер, в який будуть додаватись товари. Потім перевіряє чи кошик пустий і в випадку якщо так, виводить напис, що вказує на необхідність додати товар і посилається на каталог. Відповідний код зображено на рисунку 3.36. Далі формуються товари, а саме у вигляді: зображення, назва, ціна й ціна зі знижкою, якщо така є. Окрім цього додаються поля для зміни кількості одиниці товару й видалення товару з кошику. Для зміни кількості додається обробник події, що викликає функцію `updateItemCount()`. Для видалення додається обробник події, що викликає функцію `removeFromCart()`. Також оновлюється загальна сума замовлення, викликаючи функцію `calcTotal()`. Відповідний код зображено на рисунку 3.37. Всі згенеровані об'єкти додаються до контейнера.

```

3 usages
function loadCart() {
  const cartContainer = document.querySelector( selectors: '.items');
  cartContainer.innerHTML = '';

  if (cart.length === 0) {
    const emptyCart = document.createElement( tagName: 'h4');
    emptyCart.innerHTML = 'Ваш кошик пустий, перейдіть у <a href="catalog.html"><span class="text-span-2">каталог</span></a>, щоб додати товари.';
    cartContainer.appendChild(emptyCart);
    return;
  }
}

```

Рисунок 3.36 – Частина функції loadCart(), що відповідає за перевірку кошику на наявність товарів в cart.js

```

const countContainer = document.createElement( tagName: 'div');
countContainer.classList.add('count-container');

const countLabel = document.createElement( tagName: 'p');
countLabel.textContent = 'Кількість: ';
countLabel.classList.add('count-name');

countContainer.appendChild(countLabel);

const countInput = document.createElement( tagName: 'input');
countInput.type = 'number';
countInput.value = item.count;
countInput.min = '1';
countInput.classList.add('count-input');

countInput.addEventListener( type: 'change', listener: function (event :Event) {
  const newCount = parseInt(event.target.value);
  updateItemCount(item.id, newCount);
});

countContainer.appendChild(countInput);

const totalElement = document.querySelector( selectors: '.total-price');
const totalPrice = calcTotal();
totalElement.textContent = 'Сума: ' + totalPrice + ' грн';

const deleteButton = document.createElement( tagName: 'button');
deleteButton.textContent = '';
deleteButton.classList.add('delete-button');

deleteButton.addEventListener( type: 'click', listener: function () {
  removeFromCart(item.id);
});

itemElement.appendChild(infoElement);
itemElement.appendChild(countContainer);
itemElement.appendChild(deleteButton);
cartContainer.appendChild(itemElement);

```

Рисунок 3.37 – Частина функції loadCart(), що відповідає за змірку кількості й видалення товару в cart.js

Для зміни кількості товару створюється функція `UpdateItemCount()`. Шукається товар і якщо він є у кошику, його кількість оновлюється, а загальна ціна замовлення змінюється. Дані зберігаються у `localStorage`.

```
function updateItemCount(itemId, newCount) {
  const itemUpdate = cart.find((item) => item.id === itemId);
  if (itemUpdate) {
    itemUpdate.count = newCount;

    localStorage.setItem('cart', JSON.stringify(cart));
    updateTotal();
  }
}
```

Рисунок 3.38 – Код функції `updateItemCount` в `cart.js`

Подібним чином створюється функція для видалення товару `removeFromCart()`.

Далі створюємо функції `calcTotal()` і `updateTotal()`, що будуть відповідати за підрахування й оновлення суми замовлення. В функції `calcTotal()` створюється змінна, що накопичує ціни товарів, враховуючи, чи є на них знижка. Функція `updateTotal()` відображує цю суму у відповідному полі в реальному часі.

```
function calcTotal() {
  let total = 0;
  cart.forEach(function (item) {
    if (item.newPrice !== null)
      total += item.count * item.newPrice
    else total += item.count * item.price;
  });

  return total;
}

4 usages
function updateTotal() {
  const totalElement = document.querySelector('select: ".total-price');
  const total = calcTotal();
  totalElement.textContent = 'Сума: ' + total + ' грн';
}
```

Рисунок 3.39 – Код функцій `calcTotal()` і `updateTotal()` в `cart.js`

Для реалізації відправки замовлення на електронну форму продавця, потрібно сформуванати форму, що буде містити дані для доставки замовника, а також список товарів. Для початку, перевіряється чи кошик порожній. Якщо так – виводиться повідомлення, що потрібно додати хоча б один товар. Створюється об'єкт `formData`, в який будуть додаватися дані з форми, дані про товар й сума замовлення. Потім відправляється запит на сервер і оброблюється відповідь. В кінці кошик очищується за допомогою функції `clearCart()`.

```
const form = document.querySelector( selectors: '.order-form');

form.addEventListener( type: 'submit', listener: function (event :Event ) {
  event.preventDefault();

  if (cart.length === 0) {
    alert('Ваш кошик пустий. Додайте хоча б одну одиницю товару. ');
    return;
  }

  const formData = new FormData(form);

  cart.forEach(function (item, index) {
    formData.append( name: 'Назва товару ' + (index + 1) , item.name);
    if (item.newPrice != null) {
      formData.append( name: 'Ціна товару ' + (index + 1), item.newPrice);
    } else {
      formData.append( name: 'Ціна товару ' + (index + 1), item.price);
    }
    formData.append( name: 'Кількість товару ' + (index + 1), item.count);
  });

  const total = calcTotal();
  formData.append( name: 'Сума замовлення', value: total + ' грн');

  const xhr = new XMLHttpRequest();
  xhr.open( method: 'POST', form.action, async true);
  xhr.setRequestHeader( name: 'Accept', value: 'application/json');
  xhr.onreadystatechange = function () {
    if (xhr.readyState === XMLHttpRequest.DONE && xhr.status === 200) {
      form.reset();
      clearCart();
      alert('Дякуємо за замовлення!');
    }
  };
  xhr.send(formData);
});
```

Рисунок 3.40 – Код створення форми в `cart.js`

В кінці створюємо функцію `clearCart()`. Змінній `cart` присвоюється порожній масив. Потім з локального сховища браузера видаляється ключ `'cart'`. Викликаються функції `updateTotal()` для оновлення відображення загальної суми і `loadCart()` для перезавантаження кошика і оновлення відображення.

```
function clearCart() {  
  cart = [];  
  localStorage.removeItem( key: 'cart');  
  updateTotal();  
  loadCart();  
}
```

Рисунок 3.41 – Код функції `clearCart()` в `cart.js`

Завершальним етапом є створення файлу `cart.css`, який відповідає за розташування й дизайн елементів. Вирішено умовно поділити екран на дві частини. Зліва знаходяться товари, які розміщуються в один стовпчик, а справа форма для оформлення замовлення, яка рухається разом з гортанням сторінки. Прикладом коду `cart.css` є задання параметрів для форми.

```
.order {  
  height: 100%;  
  position: sticky;  
  top: 10vh;  
  font-size: 20px;  
  font-weight: bold;  
  margin-right: 150px;  
  width: 200px;  
}  
  
.order-form-title {  
  width: 500px;  
  margin-bottom: 50px;  
}  
  
.order-form .input, .order-form .input-text {  
  padding-top: 15px;  
  padding-bottom: 15px;  
  width: 500px;  
}  
  
.order-form .input-text {  
  padding-bottom: 50px;  
}  
  
.total-price {  
  padding-top: 50px;  
}
```

Рисунок 3.42 – Код форми в `cart.css`

У висновку, отримуємо сторінку кошику, що задовольняє всім потребам дизайну й функціональності.

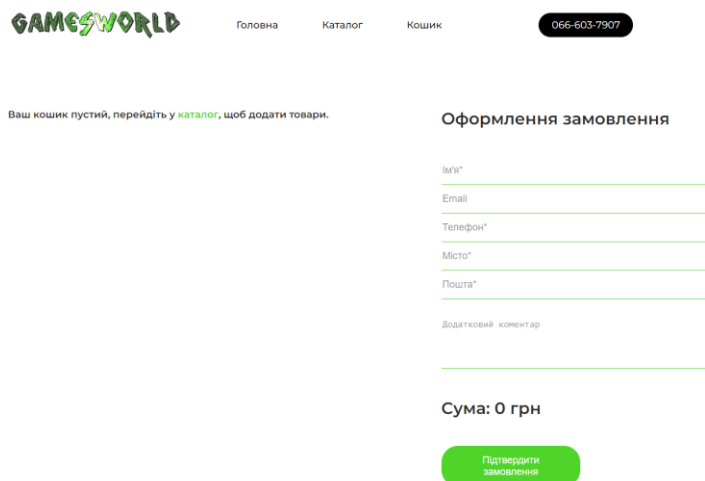


Рисунок 3.43 – Зовнішній вигляд пустого кошику

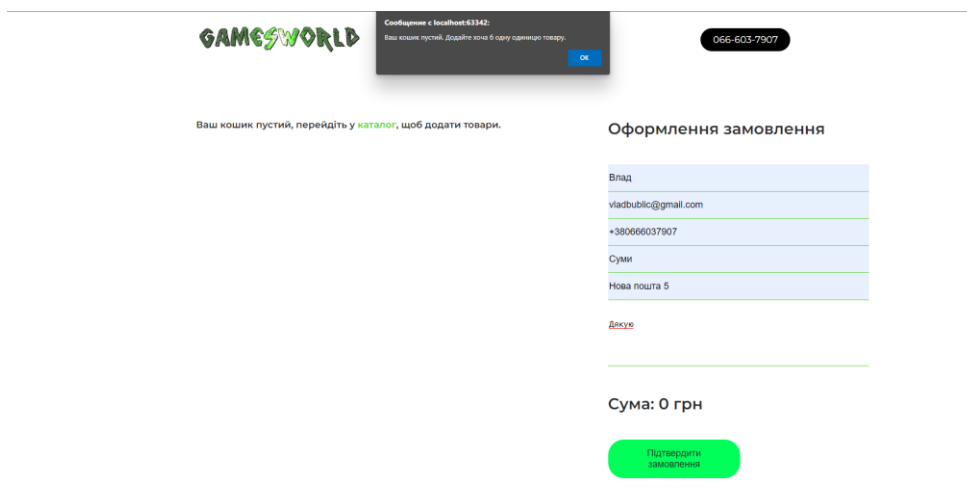


Рисунок 3.44 – Спроба підтвердження пустого замовлення

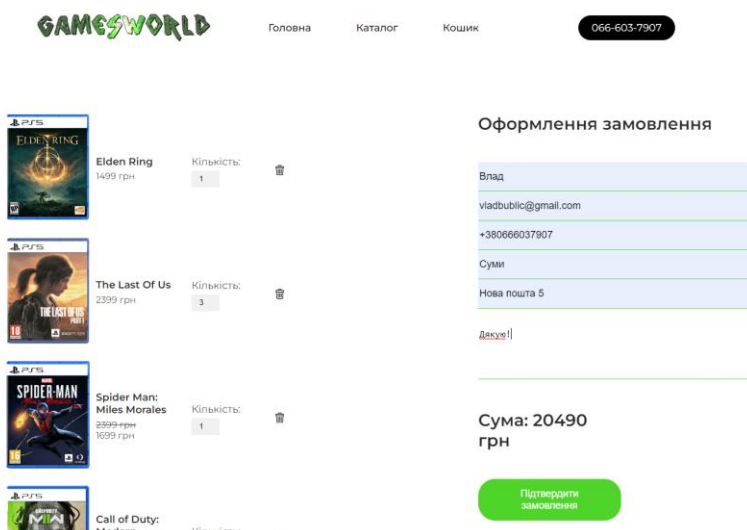


Рисунок 3.45 – Зовнішній вигляд заповненого кошику

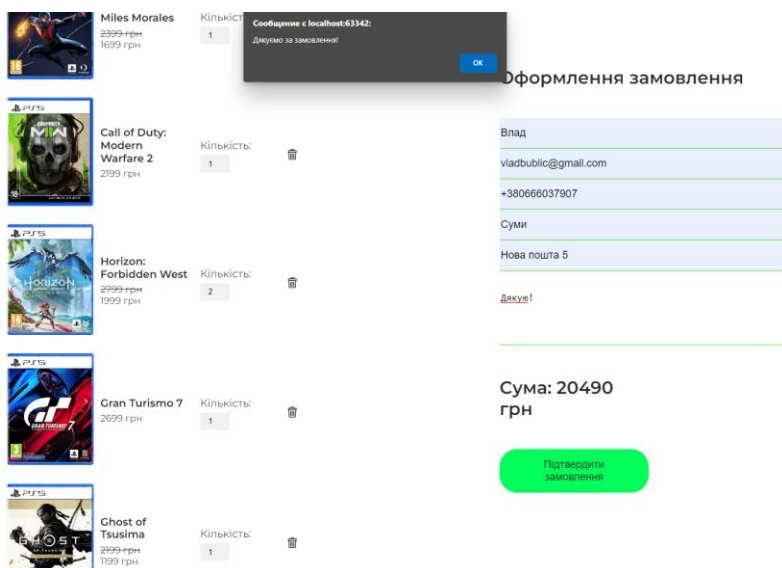


Рисунок 3.46 – Робота кнопки підтвердження замовлення, демонстрація закріплення форми на екрані

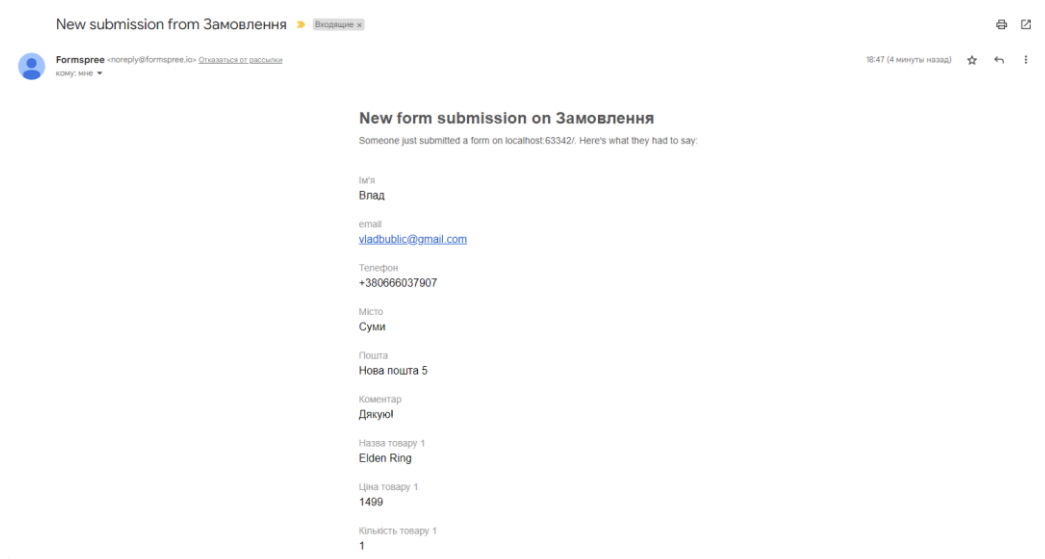


Рисунок 3.47 – Отримання даних форми замовлення на електронну пошту

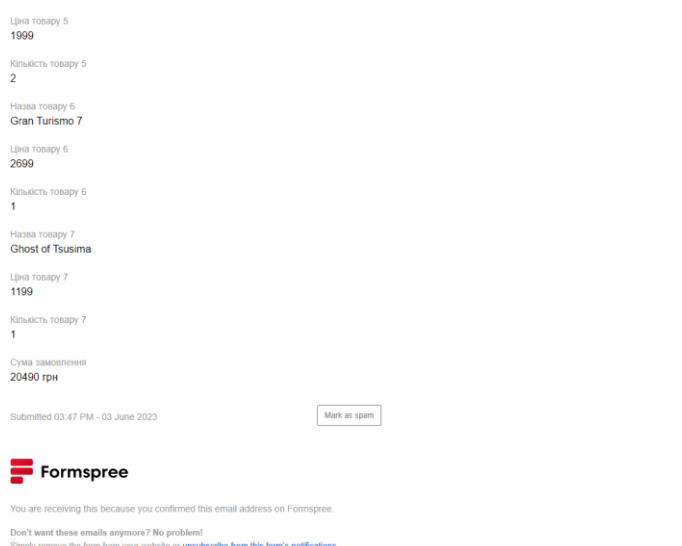


Рисунок 3.48 – Отримання даних форми замовлення на електронну пошту

ВИСНОВКИ

Результатом виконання даної кваліфікаційної роботи є створення веб-сайту "GamesWorld", що представляє собою проект інтернет-магазину відеоігор й має функціональні можливості для подальшого вдосконалення. Розглянуті та вивчені основні аспекти створення веб-сайтів. Досліджені їх основні елементи, методи розробки та етапи життєвого циклу.

Всі завдання, що описані в постановці роботи до проекту виконані. Результатами проведених досліджень у роботі є наступне:

1. Визначено тематику, потреби й цільову аудиторію інтернет магазину "GamesWorld".
2. Під час аналізу внутрішніх структур веб-сайтів були враховані особливості і потреби проекту "GamesWorld" і обрано лінійну структуру, яка найкраще відповідає потребам магазину та сприяє легкій навігації користувачів.
3. Була визначена основна логіка користування додатком та побудовані відповідні діаграми.
4. Одним із ключових етапів підготовки було визначення дизайну, кольорів та шрифтів веб-сайту. Враховуючи тематику магазину "GamesWorld", були обрані такі рішення, що сприяють створенню привабливого, гармонійного та зручного інтерфейсу.
5. Для програмної реалізації веб-сайту було вирішено використовувати такі інструменти як HTML, CSS, JavaScript та JSON. Також вивчено можливості і використано API від FormSpree для відправки форм на електронну пошту. Як середовище розробки було розглянуто та обрано PhpStorm, оскільки він надає велику кількість можливостей і зручних рішень.

В результаті отримано повноцінний, готовий до роботи, веб-сайт з дотриманням всіх поставлених потреб. Надалі планується наповнювати сайт новими товарами й підтримувати їх наявність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. HTML-підручник. W3schoolsua.українською. [Електронний ресурс] - Режим доступу:
URL:<https://w3schoolsua.github.io/html/index.html#gsc.tab=0>
2. CSS-підручник. W3schoolsua.українською. [Електронний ресурс] - Режим доступу:
URL:<https://w3schoolsua.github.io/css/index.html#gsc.tab=0>
3. JavaScript-підручник. W3schoolsua.українською. [Електронний ресурс] - Режим доступу:
URL:<https://w3schoolsua.github.io/js/index.html#gsc.tab=0>
4. Етапи створення веб-сайтів. Webtune. [Електронний ресурс] - Режим доступу:
URL:<https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/>
5. Шалева О.І. – Електронна комерція. Навчальний посібник. – К.: "Центр учбової літератури" [Електронний ресурс] - Режим доступу:
URL:https://kcollegeht.kr.ua/images/Electronni_pidruchniki/EK_Shaleva_navch_posibnik_2011p.pdf
6. Джон Дакетт. HTML та CSS. Розробка і дизайн веб-сайтів. Видавництво: Ексмо, 2016. 480с.
7. Джон Яблонські. Закони UX-дизайну. Розуміння психології користувача – ключ до успіху. Видавництво: Print2print, 2021. 160с.
8. ТОП-16 принципів створення інтернет магазину для його успішного функціонування. Education.ua. [Електронний ресурс] - Режим доступу:
URL:<https://www.education.ua/articles/664/>
9. Як будувати UML-діаграми. DOU. [Електронний ресурс] - Режим доступу:
URL:<https://dou.ua/forums/topic/40575/>

10. Інтернет-магазин. Wikipedia, the free encyclopedia. [Електронний ресурс] - Режим доступу:
URL: <https://uk.wikipedia.org/wiki/Інтернет-магазин>
11. Веб-сайт. Wikipedia, the free encyclopedia. [Електронний ресурс] - Режим доступу:
URL: <https://uk.wikipedia.org/wiki/Вебсайт>
12. Етапи створення (розробки) web-сайту. Процес створення сайтів. ITStatti. [Електронний ресурс] - Режим доступу:
URL: <https://itstatti.in.ua/stvorennya-sajtiv/115-etapi-stvorennya-rozrobki-web-sajtu-protses-stvorennya-sajtiv.html>
13. Приклад JSON stringify: як парсити об'єкт JSON за допомогою JS. freeCodeCamp. [Електронний ресурс] - Режим доступу:
URL: <https://www.freecodecamp.org/ukrainian/news/tryklad-json-stringify-yak-parsyty-ob-yekt-json-za-dopomohoyu-js/>
14. ВИПУСКНА РОБОТА на тему: «Клієнтська частина веб-сайту за допомогою JavaScript». Теницька А.С. [Електронний ресурс] - Режим доступу:
URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/80426/1/Tenytska_bac_rob.pdf
15. КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему: «Веб-сайт для продажу комп'ютерної периферії». Хомов Ю.В. [Електронний ресурс] - Режим доступу:
URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/81436/1/bachelor_thesis_Khomov.pdf

ДОДАТОК А

HTML-файли додатку

index.html

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>GamesWorld</title>
  <meta name="description" content="Games">
  <meta name="keywords" content="GamesWorld">
  <link rel="shortcut icon" href="assets/img/favicon.ico" type="image.jpg">

  <meta property="og:title" content="GamesWorld">
  <meta property="og:description" content="Videogames shop">
  <meta property="og:type" content="shop">
  <meta property="og:site_name" content="GamesWorld">

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;70
0&display=swap" rel="stylesheet">

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

</head>
<body>
<header class="header" id="header">
  <div class="container">
    <div class="header-set">
      <a class="logo">
        
      </a>
      <ul class="menu">

        <li class="menu_item">
          <a href="#" class="menu_item_text">
            Головна
          </a>
        </li>

        <li class="menu_item">
          <a href="catalog.html" class="menu_item_text">
            Каталог
          </a>
        </li>

        <li class="menu_item">
          <a href="cart.html" class="menu_item_text">
            Кошик
          </a>
        </li>
      </ul>
      <a href="tel:+380666037907" class="phone">
        066-603-7907
      </a>
    </div>
  </div>

```

```

</header>

<section class="info" id="info">
  <div class="container">
    <div class="info-set">
      <h1 class="title">
        Магазин комп'ютерних ігор
      </h1>
      <h3 class="info-title">
        Відкрийте для себе нові <span class="info-span">світи!</span>
      </h3>
      <p class="info-text">
        Найнижчі ціни
      </p>
      <a href="catalog.html" class="btn">
        Каталог
      </a>
    </div>
  </div>
</section>

<section class="about" id="about">
  <div class="container">
    <div class="about-set">
      <div class="about-text">
        <div class="about_wrap">
          Про нас
        </div>
        <h2 class="about_title">
          Ігри - це наше <span class="text-span-2">натхнення</span>
        </h2>
        <p class="about_text">
          Кожен знаходить в відеоіграх щось рідне для себе. Для
          когось це лише розвага після важкого дня, для
          когось - окремі світи.
          А для нас - це ще робота, яку ми з радістю виконуємо, щоб
          подарувати вам незабутні враження! Ми є
          невеликим приватним магазином,
          що тільки починає свій шлях, але сподіваємося, що
          принесемо вам лише позитивні емоції.
          Телефонуйте та робіть замовлення, щоб скоріше поринути у
          неймовірний віртуальний світ!
        </p>
        
      </div>
    </div>
  </div>
</section>

<section class="popular" id="popular">
  <div class="container">
    <h2 class="title-2">
      Популярні товари
    </h2>
    <p class="text-2">
      Зараз у тренді
    </p>
    <div class="item-set">
      <div class="item">
        
    <div class="item-column">
        <h4 class="item_title">
            Call Of Duty: Modern Warfare 2
        </h4>
        <p class="item_text">
            > Видавець: Activision

        </p>
        <p class="item_text">
            > Розробник: Infinity Ward, Raven Software

        </p>
        <p class="item_text">
            > Серія: Call of duty

        </p>
        <p class="item_text">
            > Жанр гри: Шутер від першої особи (FPS)
        </p>

        <a href="https://www.youtube.com/watch?v=r72GP1PIZa0"
class="item_button">Трейлер</a>
    </div>

</div>
<div class="item">
    
    <div class="item-column">
        <h4 class="item_title">
            Horizon Forbidden West
        </h4>
        <p class="item_text">
            > Видавець: Sony Interactive Entertainment

        </p>
        <p class="item_text">
            > Розробник: Guerrilla Games

        </p>
        <p class="item_text">
            > Серія: Horizon

        </p>
        <p class="item_text">
            > Жанр гри: Action
        </p>

        <a href="https://www.youtube.com/watch?v=Lq594XmpPBg"
class="item_button">Трейлер</a>
    </div>

</div>
<div class="item">
    
    <div class="item-column">
        <h4 class="item_title">
            Elden Ring

```

```

</h4>
<p class="item_text">
    > Видавець: Bandai Namco Entertainment

</p>
<p class="item_text">
    > Розробник: From Software

</p>
<p class="item_text">
    > Серія: Dark Souls

</p>
<p class="item_text">
    > Жанр гри: Рольові (RPG)

</p>
<a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ"
class="item_button">Трейлер</a>
</div>

</div>

</div>

<a href="catalog.html" class="btn-other">
    Всі товари
</a>
</div>

<section class="faq" id="faq">
    <div class="container">
        <h2 class="title-2">
            FAQ
        </h2>
        <p class="text-2">
            Магазин GamesWorld знаходиться в м.Суми , але ми працюємо -
<span
            class="text-span">по всій Україні.</span>
        </p>
        <div class="faq-set">
            <div class="set">
                <div class="pic">
                    
                <div class="faq-kit">
                    <h4 class="faq_title">
                        Час доставки
                    </h4>
                    <p class="faq_text">
                        Відправка відбувається у день замовлення,
                        якщо воно було зроблене до 17:00
                    </p>
                </div>
            </div>
        </div>
        <div class="pic">
            
        <div class="faq-kit">
            <h4 class="faq_title">

```

```

        Упаковка
    </h4>
    <p class="faq_text">
        Всі замовлення запаковуються в пузирчатку
        плівку, що забезпечує цілісність товару
    </p>
</div>

</div>
</div>
<div class="set">
    <div class="pic">
        
        <div class="faq-kit">
            <h4 class="faq_title">
                Оригінальність товару
            </h4>
            <p class="faq_text">
                Всі товари є ліцензованими та пройшли
                перевірку
            </p>
        </div>
    </div>
    <div class="pic">
        
        <div class="faq-kit">
            <h4 class="faq_title">
                Оформлення замовлення
            </h4>
            <p class="faq_text">
                Для оформлення замовлення вам достатньо
                заповнити форму або зателефонувати нам.
            </p>
        </div>
    </div>
</div>
</div>
</div>
<div class="priority" id="priority">
    <div class="container">
        <div class="priority-set">
            <div class="pic-2">
                
                <h4 class="pic-2_title">
                    Доставка по всій Україні
                </h4>
                <p class="pic-2_text">
                    Нова Пошта \ Укр Пошта \ Meest
                </p>
            </div>
        </div>
    </div>
</div>

```

```

2_img">
    
        Гарантія
    </h4>
    <p class="pic-2_text">
        Ви можете повернути товар протягом 2-х днів
    </p>
</div>
<div class="pic-2">
    
    <h4 class="pic-2_title">
        Високий рейтинг від покупців
    </h4>
    <p class="pic-2_text">
        Тільки позитивні відгуки
    </p>
</div>
</div>
</section>
</section>
<section class="call" id="call">
    <div class="container">
        <div class="order-set">
            <div class="form-wrap">
                <h4 class="form-title">
                    Замовити дзвінок
                </h4>
                <p class="form-text">
                    Заповніть поля знизу, і ми зв'яжемося з вами, щоб
                    обговорити всі питання
                </p>
                <form action="https://formspree.io/f/mayzkond" method="POST">
                    <label>
                        <input type="text" placeholder="Ім'я*" name="Ім'я"
                        class="input input-name" id="nameId"
                        required>
                        <input type="email" placeholder="Email" name="email"
                        class="input">
                        <input type="tel" placeholder="Телефон*"
                        name="Телефон" class="input" id="phoneId" required>
                        <textarea placeholder="Додатковий коментар"
                        name="Коментар" class="input input-text"></textarea>
                        <button type="submit" class="btn-form" id="btn-
                        form">Надіслати</button>
                    </label>
                </form>
            </div>
            <div class="ellipse">
                <div class="ellipse-1"></div>
                <div class="ellipse-2"></div>
                <div class="ellipse-3"></div>
                <div class="call-text">
                    <div class="call_text_wrap">
                        Приемної гри
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>

```

```

        </div>
        <p class="call_text_wrap-2">
            Ми цінуємо кожного покупця та з нетерпінням чекаємо
на ваше повернення
        </p>
    </div>
</div>
</div>
</div>
</section>

<footer class="footer" id="footer">
    <div class="container">
        <h2 class="title-3">
            Інформація про нас
        </h2>
        <p class="text-3">
            Наші контакти
        </p>
        <div class="contacts">

            <p class="contact-text">
                Бублик Владислав Олегович
            </p>

            <div class="contact-set">
                <a href="mailto:vladpublic@gmail.com" class="contact-
email">vladpublic@gmail.com</a>
                <a href="tel:0666037907" class="contact-
phone">+380666037907</a>
            </div>

            <p class="contact-text">
                Україна м.Суми
            </p>

        </div>
        <div class="social">
            <a href="https://www.facebook.com/profile.php?id=100007657141376"
class="social-item">
                
            </a>
            <a href="https://www.instagram.com/vladpublic/" class="social-
item">
                
            </a>
            <a href="https://twitter.com/VBublic" class="social-item">
                
            </a>
            <a href="https://www.linkedin.com/in/vladyslav-bublyk-3247951b9/"
class="social-item">
                
            </a>
        </div>
        <p class="text-4">
            ©GamesWorld 2023
        </p>
    </div>
</footer>
<script type="text/javascript" src="/assets/js/main.js"></script>
</body>
</html>

```


catalog.html

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>Catalog</title>
  <link rel="shortcut icon" href="assets/img/favicon.ico" type="image.jpg">

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;70
0&display=swap" rel="stylesheet">

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

<body>
<header class="header">
  <div class="container">
    <div class="header-set">
      <a class="logo">
        
      </a>
      <ul class="menu">

        <li class="menu_item">
          <a href="index.html" class="menu_item_text">
            Головна
          </a>
        </li>

        <li class="menu_item">
          <a href="#" class="menu_item_text">
            Каталог
          </a>
        </li>

        <li class="menu_item">
          <a href="cart.html" class="menu_item_text">
            Кошик
          </a>
        </li>
      </ul>
      <a href="tel:+3806666037907" class="phone">
        066-603-7907
      </a>

    </div>
  </div>
</header>

<section class="search">

  <label for="searchInput"></label>
  <input type="text" placeholder="Введіть назву товару" name="text"
class="input input-search" id="searchInput">
  <button onclick="search()" class="btn-search"></button>

</section>

```

```

<section class="catalog" id="catalog">
  <div class="items">
  </div>
</section>
<script type="text/javascript" src="/assets/js/catalog.js"></script>
</body>
</html>

```

cart.html

```

<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <title>Cart</title>
  <link rel="shortcut icon" href="assets/img/favicon.ico" type="image.jpg">

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;70
0&display=swap" rel="stylesheet">

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

</head>
<body>

<header class="header">
  <div class="container">
    <div class="header-set">
      <a class="logo">
        
      </a>
      <ul class="menu">

        <li class="menu_item">
          <a href="index.html" class="menu_item_text">
            Головна
          </a>
        </li>

        <li class="menu_item">
          <a href="catalog.html" class="menu_item_text">
            Каталог
          </a>
        </li>

        <li class="menu_item">
          <a href="#" class="menu_item_text">
            Кошик
          </a>
        </li>
      </ul>
      <a href="tel:+380666037907" class="phone">
        066-603-7907
      </a>

    </div>

```

```

    </div>
</header>

<div class="container">
  <div class="cart">
    <div class="items">
      </div>

    <div class="order">
      <form action="https://formspree.io/f/mknaerbw" method="POST"
class="order-form">
        <h2 class="order-form-title">
          Оформлення замовлення
        </h2>
        <label>
          <input type="text" placeholder="Ім'я*" name="Ім'я"
class="input" id="nameId" required>
          <input type="email" placeholder="Email" name="email"
class="input">
          <input type="tel" placeholder="Телефон*" name="Телефон"
class="input" id="phoneId" required>
          <input type="text" placeholder="Місто*" name="Місто"
class="input" id="cityId" required>
          <input type="text" placeholder="Пошта*" name="Пошта"
class="input" id="postId" required>
          <textarea placeholder="Додатковий коментар"
name="Коментар" class="input input-text"></textarea>

        </label>
        <h2 class="total-price">

        </h2>

        <button type="submit" class="checkout-btn">Підтвердити
замовлення</button>
      </form>

    </div>
  </div>

</div>
<script type="text/javascript" src="/assets/js/cart.js"></script>
</body>
</html>

```

ДОДАТОК Б

CSS файлы проекту

main.css

```
body {
  font-family: 'Montserrat', sans-serif;
  font-size: 18px;
  color: rgba(42, 42, 42, 0.9);
  margin: 0;
  font-weight: 400;
  overflow-x: hidden;
}

:active, :hover, :focus {
  outline: 0;
  outline-offset: 0;
}

* {
  box-sizing: border-box;
  text-decoration: none;
}

h1, h2, h3, h4, h5, h6 {
  padding: 0;
  margin: 0;
  font-weight: 600;
  line-height: 1.2em;
  color: #2A2A2A;
}

img {
  display: block;
  max-width: 100%;
}

ul, li {
  display: block;
  padding: 0;
  margin: 0;
}

p {
  margin: 0;
}

.container {
  width: 1140px;
  margin: 0 auto;
}

.header {
  padding: 23px 0;
}

.header-set {
```

```
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.menu {
    display: flex;
}

.menu_item_text {
    font-weight: 500;
    font-size: 18px;
    line-height: 22px;
    margin-right: 80px;
    color: #000000;
}

.phone {
    background-color: #000000;
    color: #ffffff;
    padding: 11px 24px;
    border-radius: 25px;
}

.info {
    background-image: url(https://wallpapercave.com/wp/x6Z3YLM.jpg);
    background-color: #cccccc;
    padding-bottom: 100px;
}

.title {
    padding: 123px 0 30px;
    font-weight: 700;
    font-size: 55px;
    width: 643px;
    color: #FFFFFF;
}

.info-title {
    font-weight: 600;
    font-size: 33px;
    line-height: 40px;

    color: #FFFFFF;
}

.info-span {
    color: #4fd529;
}

.info-text {
    color: rgba(255, 255, 255, 0.9);
    padding: 30px 0;
}

.btn {
    display: block;
    background: #4fd529;
    border-radius: 30px;
    color: #ffffff;
    padding: 17px 30px;
    width: 130px;
}
```

```
}

.btn:hover {
  color: #000000;
  background: #00ff59;
}

.about {
  padding: 125px 0;
}

.about-set {
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.about_wrap {
  width: 100px;
  text-align: center;
  padding: 5px 0;
  color: rgba(255, 255, 255, 0.9);
  background: #2f3648;
  border-radius: 30px;
}

.about_title {
  font-weight: 600;
  font-size: 40px;
  width: 410px;
  color: #2A2A2A;
  padding: 15px 0 30px;
}

.text-span-2 {
  color: #4fd529;
}

.about_text {
  line-height: 1.5em;
  width: 560px;
  color: rgba(42, 42, 42, 0.9);
}

.about_img {
  margin-right: -100px;
}

.popular {
  margin-left: auto;
  margin-right: auto;
}

.item-set {
  text-align: center;
  display: flex;
  justify-content: center;
}

.item {
  align-items: center;
  display: flex;
}
```

```

        flex-direction: column;
    }

    .item-column {
        width: 360px;
        height: 500px;
        background: #FFFFFF;
    }

    .item-img {
        margin-bottom: 30px;
        width: auto;
        height: 400px;
    }

    .item_title {
        text-align: center;
        padding-top: 30px;
        font-weight: 600;
        font-size: 22px;

        color: #2A2A2A;
    }

    .item_text {
        width: 260px;
        margin: 25px auto;
        text-align: center;
        font-size: 18px;
        color: rgba(42, 42, 42, 0.9);
    }

    .item_button {
        font-weight: 500;
        font-size: 18px;
        padding: 15px 41px;
        display: block;
        background: linear-gradient(90deg, #4fd529 0%, #00ff59 100%);
        border-radius: 30px;
        color: #ffffff;
        text-align: center;
        margin: 30px auto;
    }

    .item-button-span-white {
        color: #ffffff55;
        text-decoration-line: line-through;
    }

    .item_button:hover {
        background: #00ff59;
    }

    .item-button-mod {
        font-weight: 500;
        font-size: 18px;
        padding-top: 7px;
        padding-bottom: 8px;
        display: grid;
        grid-template-columns: 1fr;
        grid-template-rows: 50% 50%;
    }

```

```
border-radius: 30px;
text-align: center;
margin: 30px auto;
color: #fff;
background: linear-gradient(90deg, #ffcc00 0%, #FCE043 100%);
}

.item_button, .item-button-mod {
width: 80%;
height: 3em !important;
}

.item-button-mod :first-child {
font-size: 0.8em;
}

.item-button-mod:hover {
background: #FCE043;
}

.btn-other {
color: #000000;
display: flex;
justify-content: center;
align-items: center;
text-decoration: underline;
}

.btn-other:hover {
color: #7e7e7e;
}

.faq {
background: #ffffff;
padding: 100px 0;
}

.title-2 {
text-align: center;
font-weight: 600;
font-size: 40px;
line-height: 49px;

color: #4fd529;
}

.text-2 {
width: 483px;
text-align: center;
margin: 30px auto 60px;
}

.text-span {
font-weight: 600;
}

.faq-set {
display: flex;
justify-content: space-between;
```



```
}  
  
.set {  
  margin-bottom: 30px;  
}  
  
.pic {  
  display: flex;  
  margin-bottom: 40px;  
}  
  
.pic-img {  
  width: 44px;  
  height: 44px;  
  margin-right: 17px;  
}  
  
.faq-kit {  
  width: 440px;  
}  
  
.faq_title {  
  margin: 10px 0 30px;  
}  
  
.btn-call {  
  width: max-content;  
  font-weight: 500;  
  padding: 15px 95px;  
  background: #4fd529;  
  color: #fff;  
  border-radius: 30px;  
  display: block;  
  margin: 30px auto 0;  
}  
  
.btn-call:hover {  
  color: #262626;  
  background: #00ff59;  
}  
  
.priority {  
  padding: 100px 0;  
}  
  
.priority-set {  
  display: flex;  
  justify-content: space-between;  
}  
  
.pic-2 {  
  width: 300px;  
}  
  
.pic-2_img {  
  margin: 0 auto;  
}  
  
.pic-2_title {  
  margin: 44px 0;  
  font-weight: 600;  
}
```

```
    font-size: 22px;
    text-align: center;
}

.pic-2_text {
    color: rgba(42, 42, 42, 0.9);
    line-height: 1.4em;
    text-align: center;
}

.call {
    background: url("https://digwallpapers.com/wallpapers/full/0/3/a/82441-1920x1080-retro-game-background-image-desktop-hd.jpg");
    padding: 100px 0;
}

.order-set {
    display: flex;
    justify-content: space-between;
}

.form-wrap {
    background: #FFFFFF;
    border-radius: 30px;
    width: 600px;
    padding: 60px;
    height: 630px;
}

.form-text {
    width: 440px;
    color: rgba(42, 42, 42, 0.9);
    margin: 13px 0 23px;
}

.form-title {
    font-weight: 600;
    font-size: 22px;
    width: 500px;
    color: #2A2A2A;
}

input::placeholder {
    font-weight: 500;
    font-size: 18px;
    color: #2a2a2a88;
}

.input {
    padding-bottom: 5px;
    width: 225px;
    border: none;
    border-bottom: 1px solid #4fd529;
    font-size: 18px;
}

.input-name {
    width: 470px;
    margin-bottom: 30px;
}

.label {
```

```
        color: rgba(42, 42, 42, 0.9);
    }

    .input-text {
        width: 474px;
        font-weight: 500;
        font-size: 17px;
        margin-top: 20px;
        border: none;
        border-bottom: 1px solid #4fd529;
        resize: none;
        padding-bottom: 100px;
    }

    textarea::placeholder {
        color: #2a2a2a88;
        font-size: 17px;
        font-weight: 500;
    }

    .btn-form {
        font-weight: 500;
        font-size: 18px;
        margin-top: 50px;
        background: linear-gradient(90deg, #4fd529 0%, #00ff59 100%);
        border-radius: 30px;
        padding: 16px 185px;
        width: 100%;
        border: none;
        display: block;
        color: #ffffff;
        cursor: pointer;
    }

    .btn-form:hover {
        font-weight: 500;
        font-size: 18px;
        margin-top: 50px;
        background: #00ff59;
        border-radius: 30px;
        padding: 16px 185px;
        width: 100%;
        border: none;
        display: block;
        color: #262626;
        cursor: pointer;
    }

    .ellipse{
        display: flex;
        flex-direction: column;
        padding-top: 40px;
        margin-bottom: 20px;
    }

    .ellipse-1{
        width: 87px;
        height: 87px;
        background: #FCE043;
        border-radius: 50%;
    }

    .ellipse-2{
```

```
    width: 197px;
    height: 197px;
    background: #FCE043;
    border-radius: 50%;
    margin-left: 80px;
    margin-top: -40px;
}
.ellipse-3{
    width: 69px;
    height: 69px;
    background: #FCE043;
    border-radius: 50%;
    margin-left: 150px;
    margin-top: 60px;
}

.call_text_wrap {
    width: 180px;
    text-align: center;
    background: #2A2A2A;
    border-radius: 30px;
    margin: 74px 0 20px;
    color: rgba(239, 239, 239, 0.9);
}

.call_text_wrap-2 {
    display: inline-block;
    height: 100px;
    font-weight: 400;
    width: 450px;
    font-size: 22px;
    color: #000000;
    background: #ffffff;
    border-radius: 15px;
    border-left: white solid 15px;
    border-top: white solid 10px;
}

.footer {
    background: #2A2A2A;
    padding: 100px 0 30px;
}

.title-3 {
    text-align: center;
    font-weight: 600;
    font-size: 40px;
    color: #fff;
}

.text-3 {
    width: 100%;
    text-align: center;
    margin: 30px auto 60px;
    color: rgba(255, 255, 255, 0.9);
}

.contacts {
    padding: 0 100px 60px;
    display: flex;
    justify-content: space-between;
```

```
}  
  
.contact-text {  
  width: 133px;  
  color: #fff;  
  text-align: center;  
  line-height: 1.5em;  
}  
  
.contact-set {  
  display: flex;  
  flex-direction: column;  
  gap: 5px;  
}  
  
.contact-email {  
  text-align: center;  
  color: #4fd529;  
}  
  
.contact-phone {  
  color: #4fd529;  
  text-align: center;  
}  
  
.social {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}  
  
.social-item {  
  margin-right: 20px;  
}  
  
.text-4 {  
  padding-top: 50px;  
  color: #fff;  
  text-align: center;  
}
```

catalog.css

```
.search {  
  text-align: center;  
}  
  
.input-search {  
  min-width: 1030px;  
  height: 50px;  
  margin-top: 20px;  
}  
  
.btn-search {  
  margin-left: 10px;  
  background-color: transparent;  
  background-image: url('/assets/img/search.png');  
  background-repeat: no-repeat;
```

```
        background-position: center;
        border: none;
        cursor: pointer;
        background-size: 30px;
        padding: 20px;
    }

    .items {
        display: flex;
        flex-wrap: wrap;
        margin-top: 50px;
        flex-direction: row;
        gap: 20px;
    }

    .item {
        text-align: center;
        width: 280px;
        height: 430px;
        position: relative;
    }

    .item .item-img {
        margin-bottom: 0;
        width: auto;
        height: 230px;
    }

    .item .item_button, .item .item-button-mod{
        position: absolute;
        bottom: 0;
        width: 200px;
        justify-content: center;
        align-items: center;
    }

    .catalog {
        padding-top: 50px;
        padding-bottom: 50px;
        margin-left: 150px;
        justify-content: center;
        justify-items: center;
        justify-self: center;
    }
}
```

cart.css

```
.cart {
    display: flex;
    margin-top: 100px;
    justify-content: space-between;
}

.item {
    display: flex;
    align-items: center;
    width: 200px;
    flex-direction: row;
}
}
```

```
.item-img {
  width: 150px;
  height: auto;
  margin-right: 10px;
  margin-left: -50px;
}

.item-info {
  display: flex;
  flex-direction: column;
  justify-content: center;
  text-align: left;
  margin-top: -50px;
}

.item-info .item_title {
  margin-bottom: 5px;
  font-size: 18px;
  width: 150px;
  text-align: left;
}

.item-info .item_price {
  margin-bottom: 5px;
  font-size: 16px;
  width: 100px;
}

.count-container {
  display: flex;
  flex-direction: column;
  text-align: left;
  margin-top: -15px;
  margin-left: 20px;
}

.count-input {
  margin-top: 5px;
  padding: 5px;
  height: 30px;
  width: 50px;
  text-align: center;
  border: none;
  border-radius: 4px;
  background-color: #f1f1f1;
}

.delete-button {
  margin-left: 50px;
  display: flex;
  padding: 20px;
  background-color: transparent;
  border: none;
  cursor: pointer;
  background-image: url('/assets/img/trash.png');
  background-repeat: no-repeat;
  background-position: center;
  background-size: 18px;
  margin-top: -20px;
}
```

```
.order {
  height: 100%;
  position: sticky;
  top: 10vh;
  font-size: 20px;
  font-weight: bold;
  margin-right: 150px;
  width: 200px;
}

.order-form-title {
  width: 500px;
  margin-bottom: 50px;
}

.order-form .input, .order-form .input-text {
  padding-top: 15px;
  padding-bottom: 15px;
  width: 500px;
}

.order-form .input-text {
  padding-bottom: 50px;
}

.total-price {
  padding-top: 50px;
}

.checkout-btn {
  font-weight: 500;
  margin-top: 50px;
  font-size: 18px;
  border: none;
  padding: 16px 75px;
  background: #4fd529;
  color: #fff;
  border-radius: 30px;
  outline: transparent;
  cursor: pointer;
}

.checkout-btn:hover {
  color: #262626;
  background: #00ff59;
}
```


ДОДАТОК В

JavaScript файли проекту

main.js

```
const form = document.querySelector('form');
const submitButton = document.querySelector('#btn-form');

form.addEventListener('submit', function (event) {
    event.preventDefault();

    const formData = new FormData(form);

    const xhr = new XMLHttpRequest();
    xhr.open('POST', form.action, true);
    xhr.setRequestHeader('Accept', 'application/json');
    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                form.reset();
                alert('Чекайте на дзвінок!');
            } else {
                alert('Вибачте. Сталася помилка');
            }
        }
    };
    xhr.send(formData);
});
```

catalog.js

```
window.addEventListener('load', loadItemsFromJSON);

let catalog;

function loadItemsFromJSON() {
    const jsonFile = 'assets/catalog.json';

    const xhr = new XMLHttpRequest();
    xhr.overrideMimeType('application/json');
    xhr.open('GET', jsonFile, true);
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            catalog = JSON.parse(xhr.responseText);
            loadCatalog(catalog);
        }
    };
    xhr.send();
}

function search() {
    const search = document.getElementById('searchInput').value;
    let searchResult = [];
    catalog.forEach(function (item) {
        if (item.name.toLowerCase().includes(search.toLowerCase())) {
            searchResult.push(item);
        }
    });
}
```

```

    });
    if (searchResult.length !== 0)
        loadCatalog(searchResult);
    else
        alert("Товар не знайдено")
}

function loadCatalog(items) {
    const itemsContainer = document.querySelector('.items');
    itemsContainer.replaceChildren();
    items.forEach(function (item) {
        const itemElement = document.createElement('div');
        itemElement.classList.add('item');

        const imgElement = document.createElement('img');
        imgElement.src = item.image;
        imgElement.alt = '';
        imgElement.classList.add('item-img');
        itemElement.appendChild(imgElement);

        const titleElement = document.createElement('h4');
        titleElement.classList.add('item_title');
        titleElement.id = item.id;
        titleElement.textContent = item.name;
        itemElement.appendChild(titleElement);

        const buttonElement = document.createElement('a');
        buttonElement.href = item.orderLink;
        if (item.newPrice == null) {
            buttonElement.classList.add('item_button');
            buttonElement.textContent = item.price + ' грн';
        } else {
            buttonElement.classList.add('item-button-mod');
            const newPrice = document.createElement('span');
            newPrice.textContent = item.newPrice + ' грн';
            const price = document.createElement('span');
            price.classList.add('item-button-span-white');
            price.textContent = item.price + ' грн';

            buttonElement.appendChild(price);
            buttonElement.appendChild(newPrice);
        }

        buttonElement.addEventListener('click', function (event) {
            event.preventDefault();
            addToCart(item);
        });

        itemElement.appendChild(buttonElement);

        itemsContainer.appendChild(itemElement);
    });
}

function addToCart(item) {
    let cart = JSON.parse(localStorage.getItem('cart')) || [];
    console.log(cart);
    const existingItem = cart.find((cartItem) => cartItem.id === item.id);
    console.log(existingItem);
    if (existingItem) {
        existingItem.count += 1;
    }
}

```

```

    } else {
      item.count = 1;
      cart.push(item);
    }
    console.log(cart);
    localStorage.setItem('cart', JSON.stringify(cart));
    alert("Товар додано до кошику!");
  }
}

```

cart.js

```

window.addEventListener('load', loadCartFromJSON);
window.addEventListener('load', updateTotal);

let cart;

function loadCartFromJSON() {
  cart = JSON.parse(localStorage.getItem('cart')) || [];
  loadCart();
}

function loadCart() {
  const cartContainer = document.querySelector('.items');
  cartContainer.innerHTML = '';

  if (cart.length === 0) {
    const emptyCart = document.createElement('h4');
    emptyCart.innerHTML = 'Ваш кошик пустий, перейдіть у <a href="catalog.html"><span class="text-span-2">каталог</span></a>, щоб додати товари.';
    cartContainer.appendChild(emptyCart);
    return;
  }

  cart.forEach(function (item) {
    const itemElement = document.createElement('div');
    itemElement.classList.add('item');

    const imgElement = document.createElement('img');
    imgElement.src = item.image;
    imgElement.alt = '';
    imgElement.className = 'item-img';
    itemElement.appendChild(imgElement);

    const infoElement = document.createElement('div');
    infoElement.classList.add('item-info');

    const titleElement = document.createElement('h4');
    titleElement.className = 'item_title';
    titleElement.id = item.id;
    titleElement.textContent = item.name;
    infoElement.appendChild(titleElement);

    const priceElement = document.createElement('p');
    priceElement.className = 'item_price';
    if (item.newPrice == null) {
      priceElement.textContent = item.price + ' грн';
    } else {
      const priceSpan = document.createElement('span');
      priceSpan.textContent = item.price + ' грн';
    }
  });
}

```

```

        priceSpan.style.textDecoration = 'line-through';

        const newPriceSpan = document.createElement('span');
        newPriceSpan.textContent = item.newPrice + ' грн';

        priceElement.appendChild(priceSpan);
        priceElement.appendChild(document.createElement('br'));
        priceElement.appendChild(newPriceSpan);
    }

    infoElement.appendChild(priceElement);

    const countContainer = document.createElement('div');
    countContainer.classList.add('count-container');

    const countLabel = document.createElement('p');
    countLabel.textContent = 'Кількість:';
    countLabel.classList.add('count-name');

    countContainer.appendChild(countLabel);

    const countInput = document.createElement('input');
    countInput.type = 'number';
    countInput.value = item.count;
    countInput.min = '1';
    countInput.classList.add('count-input');

    countInput.addEventListener('change', function (event) {
        const newCount = parseInt(event.target.value);
        updateItemCount(item.id, newCount);
    });

    countContainer.appendChild(countInput);

    const totalElement = document.querySelector('.total-price');
    const totalPrice = calcTotal();
    totalElement.textContent = 'Сума: ' + totalPrice + ' грн';

    const deleteButton = document.createElement('button');
    deleteButton.textContent = '';
    deleteButton.classList.add('delete-button');

    deleteButton.addEventListener('click', function () {
        removeFromCart(item.id);
    });

    itemElement.appendChild(infoElement);
    itemElement.appendChild(countContainer);
    itemElement.appendChild(deleteButton);
    cartContainer.appendChild(itemElement);
});
}

function updateItemCount(itemId, newCount) {
    const itemUpdate = cart.find((item) => item.id === itemId);
    if (itemUpdate) {
        itemUpdate.count = newCount;

        localStorage.setItem('cart', JSON.stringify(cart));
        updateTotal();
    }
}
}

```

```

function removeFromCart(itemId) {
    cart = cart.filter((item) => item.id !== itemId);
    localStorage.setItem('cart', JSON.stringify(cart));
    updateTotal();
    loadCart();
}

function calcTotal() {
    let total = 0;
    cart.forEach(function (item) {
        if (item.newPrice !== null)
            total += item.count * item.newPrice
        else total += item.count * item.price;
    });

    return total;
}

function updateTotal() {
    const totalElement = document.querySelector('.total-price');
    const total = calcTotal();
    totalElement.textContent = 'Сума: ' + total + ' грн';
}

const form = document.querySelector('.order-form');

form.addEventListener('submit', function (event) {
    event.preventDefault();

    if (cart.length === 0) {
        alert('Ваш кошик пустий. Додайте хоча б одну одиницю товару.');
```

```
function clearCart() {  
  cart = [];  
  localStorage.removeItem('cart');  
  updateTotal();  
  loadCart();  
}
```

ДОДАТОК Г

JSON файл

catalog.json

```
[
  {
    "image": "assets/img/img_callofduty.png",
    "name": "Call of Duty: Modern Warfare 2",
    "id": 1,
    "price": "2199"
  },
  {
    "image": "assets/img/img_horizon.png",
    "name": "Horizon: Forbidden West",
    "id": 2,
    "price": 2799,
    "newPrice": 1999
  },
  {
    "image": "assets/img/img_eldenring.png",
    "name": "Elden Ring",
    "id": 3,
    "price": "1499"
  },
  {
    "image": "assets/img/thelastofus.jpg",
    "name": "The Last Of Us",
    "id": 4,
    "price": "2399"
  },
  {
    "image": "assets/img/spiderman.jpg",
    "name": "Spider Man: Miles Morales",
    "id": 5,
    "price": 2399,
    "newPrice": 1699
  },
  {
    "image": "assets/img/godofwar.jpg",
    "name": "God Of War: Ragnarok",
    "id": 6,
    "price": "2599"
  },
  {
    "image": "assets/img/hogwarts.jpg",
    "name": "Hogwarts Legacy",
    "id": 7,
    "price": "2499"
  },
  {
    "image": "assets/img/grantursimo.jpg",
    "name": "Gran Turismo 7",
    "id": 8,
    "price": "2699"
  },
  {
    "image": "assets/img/ghostoftsusima.jpg",
    "name": "Ghost of Tsusima",
    "id": 9,

```

```
    "price": "2199",
    "newPrice": "1199"
  },
  {
    "image": "assets/img/returnal.jpg",
    "name": "Returnal",
    "id": 10,
    "price": "2299"
  },
  {
    "image": "assets/img/deadisland.jpg",
    "name": "Dead Island 2",
    "id": 11,
    "price": "2399"
  },
  {
    "image": "assets/img/wwe.jpg",
    "name": "WWE2K23",
    "id": 12,
    "price": "2599",
    "newPrice": "2199"
  },
  {
    "image": "assets/img/uncharted.jpg",
    "name": "Uncharted",
    "id": 13,
    "price": "1799",
    "newPrice": "1499"
  },
  {
    "image": "assets/img/resident.jpg",
    "name": "Resident Evil: Village",
    "id": 14,
    "price": "2399"
  }
]
```