МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «УНІВЕРСИТЕТ «КРОК»
Фаховий коледж Університету «КРОК»

ДИПЛОМНА РОБОТА
за темою
«Розробка та створення сайту магазину іграшок»

Студент 4 курсу групи ІПЗ-20к/2 **Керівник дипломної роботи**

(посада керівника)

Міщанин Костянтин Васильович — (прізвище, ім'я та по-батькові студента)

Кириченко Віктор Вікторович — (прізвище, ім'я та по-батькові керівника)

До захисту

(резолюція «До захисту»)

(підпис студента)	10.06.2024 (дата)	(підпис викладача)

Київ, 2024 рік

АНОТАЦІЯ

Дипломна робота присвячена розробці веб-сайту для магазину іграшок за допомогою системи управління веб-контентом. В роботі досліджуються теоретичні основи систем управління контентом, аналізуються існуючі рішення на ринку іграшок та визначаються критерії вибору оптимальної CMS для задач конкретного бізнесу. В практичній частині представлено проектування архітектури сайту, розробка користувацького інтерфейсу, інтеграція з обраною системою управління контентом, а також тестування і оптимізація готового рішення. Робота має на меті підвищити ефективність онлайн-продажів та забезпечити зручність користувачів при виборі та покупці іграшок.

Ключові слова: веб-сайт, магазин іграшок, система управління контентом, веб-розробка, користувацький інтерфейс, тестування сайту, оптимізація веб-ресурсів, аналіз ринку, CMS, електронна комерція.

ЗМІСТ

ВСТУП

Сучасний світ неможливо уявити без інтернету та його можливостей, зокрема, без електронної комерції, яка з кожним роком все більше зміцнює свої позиції у світовій економіці. Веб-сайти стають важливим інструментом для бізнесу, що дозволяє залучати нових клієнтів, збільшувати продажі та вдосконалювати обслуговування клієнтів. Однією з активно розвиваючихся ніш є ринок іграшок, де конкуренція змушує компанії вдосконалювати не тільки асортимент продукції, але й способи її презентації та продажу через інтернет.

Актуальність даної роботи визначається швидким розвитком ринку електронної комерції та необхідністю впровадження ефективних інструментів для створення та управління веб-сайтами, зокрема систем управління контентом (CMS), які дозволяють оптимізувати робочі процеси і підвищити залученість користувачів.

Основною метою цієї дипломної роботи є розробка веб-сайту для магазину іграшок з використанням сучасних систем управління контентом, щоб забезпечити зручність управління контентом, ефективність обслуговування клієнтів та збільшення продажів. Задачами дослідження є: аналіз ринку веб-магазинів іграшок, вибір та аналіз функціональних можливостей існуючих CMS, проектування архітектури сайту, розробка користувацького інтерфейсу, інтеграція з CMS та тестування готового рішення.

Об'єктом дослідження є процес створення веб-сайту для магазину іграшок, а предметом – системи управління контентом, як інструмент розробки та оптимізації веб-сайту.

У висновку вступу можна сказати, що успіх сучасного бізнесу значною мірою залежить від ефективності його присутності в інтернеті. Розробка веб-

сайту з урахуванням сучасних вимог та тенденцій ринку може значно покращити позиції магазину іграшок у конкурентній боротьбі, залучаючи нових клієнтів і забезпечуючи високий рівень задоволення потреб існуючих. Використання систем управління контентом (CMS) в процесі створення та управління сайтом стає ключовим фактором у забезпеченні оперативності, безпеки та ефективності електронної комерції.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ УПРАВЛІННЯ ВЕБ-КОНТЕНТОМ

1.1 Види систем управління веб-контентом (CMS)

Системи управління веб-контентом (CMS) становлять основу сучасного веб-розробництва, оскільки дозволяють користувачам без глибоких технічних знань ефективно управляти контентом сайтів. Наявність різноманітних CMS сприяє широкому вибору платформ залежно від специфіки та потреб проекту. Розглянемо основні типи CMS, їх призначення та особливості.

Універсальні системи управління контентом

Універсальні системи управління контентом, такі як WordPress, Joomla, та Drupal, є високоадаптивними платформами, які можна застосувати для розробки різноманітних типів веб-сайтів. Ці системи забезпечують потужний функціонал, який може бути легко адаптований для персоналізованих потреб користувачів, від особистих блогів до складних корпоративних порталів.

WordPress є однією з найпопулярніших CMS у світі, завдяки своїй гнучкості та величезній спільноті[1]. WordPress пропонує тисячі тем оформлення та плагінів, які дозволяють розширити функціональність сайту без потреби в глибоких знаннях програмування. Це робить його ідеальним вибором для блогерів, малого та середнього бізнесу.

Joomla володіє більшою гнучкістю у порівнянні з WordPress і використовується для створення складніших сайтів. Вона пропонує розширені можливості управління користувачами та контентом, що є корисним для сайтів з багатошаровою структурою доступу та великою кількістю контенту, як-от новинні портали.

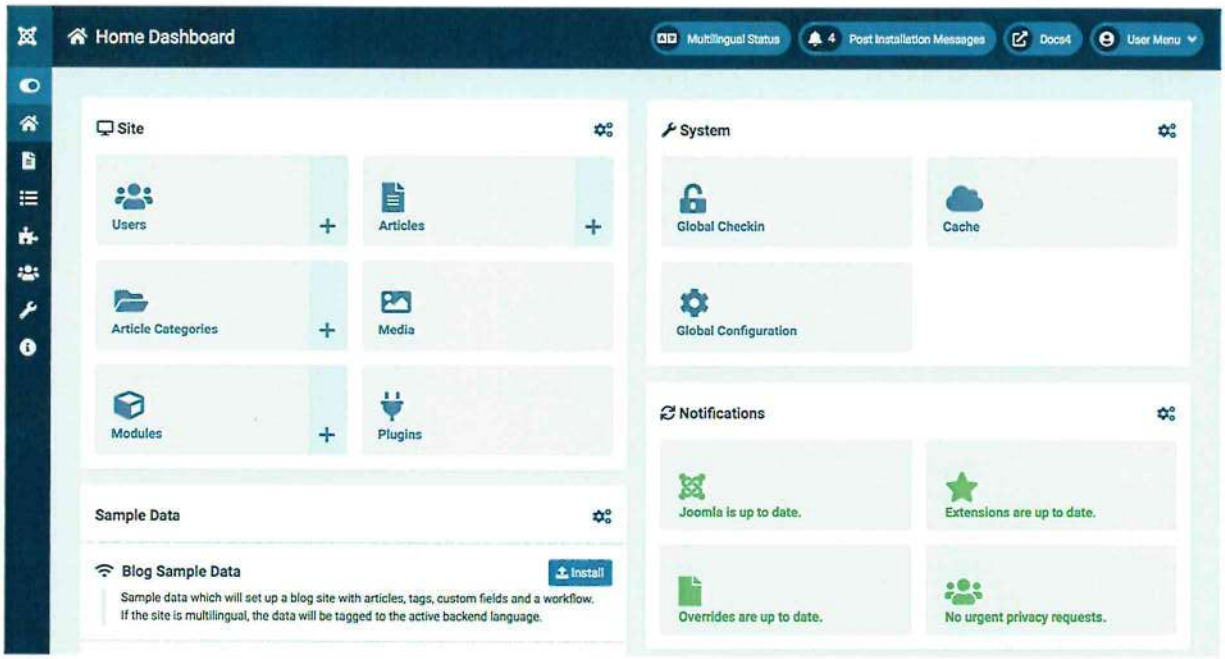


Рисунок 1.1 – Joomla CMS

Drupal є найбільш потужною серед наведених CMS і часто використовується для великих корпоративних та урядових веб-сайтів. Drupal відомий своєю високою безпекою, широкими можливостями інтеграції та масштабування, що робить його відмінним вибором для складних проектів, які вимагають високого рівня індивідуалізації.

Спеціалізовані системи управління контентом

Спеціалізовані CMS, такі як Magento та Shopify, оптимізовані для специфічних потреб електронної комерції, що робить їх ідеальним вибором для бізнесів, які зосереджені на продажу товарів через інтернет. Ці системи надають потужні інструменти для управління інвентарем, обробки замовлень та ведення клієнтської бази[2].

Magento є однією з найбільш комплексних платформ для електронної комерції, що дозволяє налаштувати кожен аспект магазину. Вона включає

обширні функції для управління товарами, кампаніями, маркетингом, і забезпечує високий рівень масштабування для зростаючих бізнесів.

Shopify пропонує більш просте та зрозуміле рішення для запуску онлайн-магазину[3]. Завдяки своїй інтуїтивно зрозумілій панелі управління та широкому набору шаблонів, Shopify став популярним вибором серед невеликих магазинів, які прагнуть швидко розпочати продажі в інтернеті.

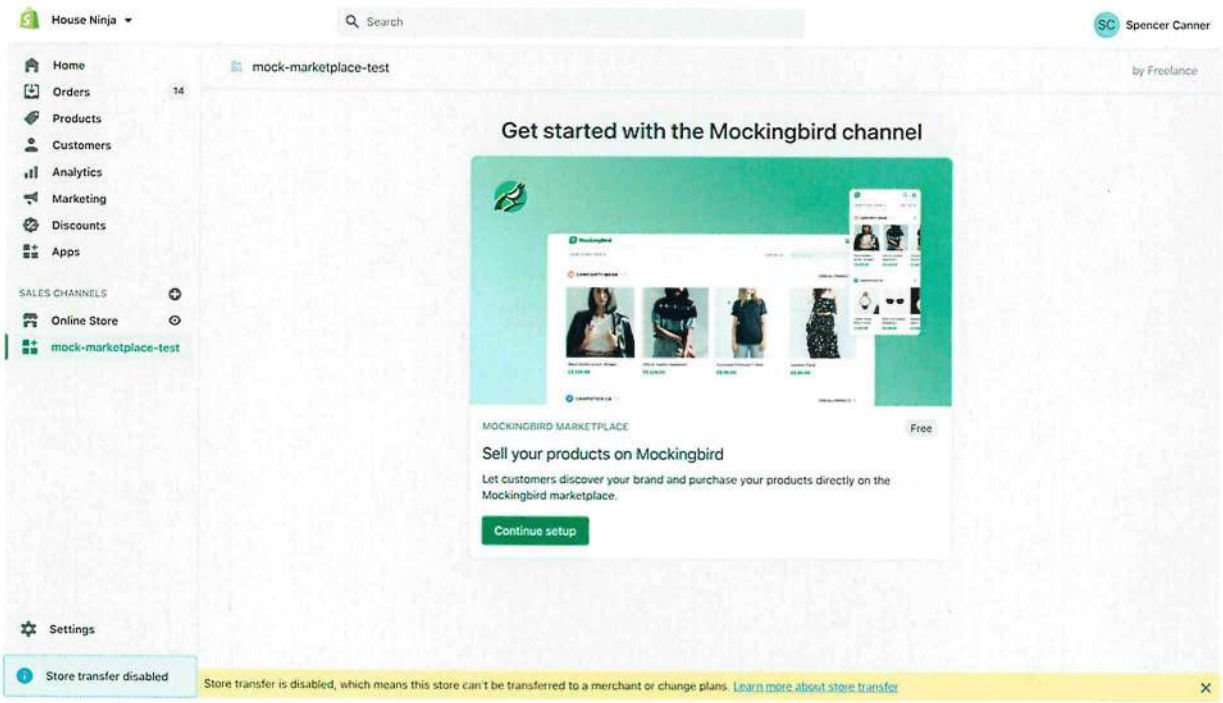


Рисунок 1.2 – Shopify

Alfresco використовується переважно великими організаціями для управління цифровим контентом, включаючи документи, електронні таблиці та медіафайли. Ця система є важливим інструментом для управління корпоративними ресурсами, підтримки співпраці в команді та забезпечення дотримання нормативних вимог.

Ці системи демонструють, як CMS можуть бути адаптовані до різних потреб бізнесу, забезпечуючи ефективне управління контентом та взаємодію з клієнтами.

CMS з відкритим кодом та комерційні CMS

Системи управління контентом з відкритим кодом та комерційні CMS становлять два основних підходи до розгортання та використання платформ для управління веб-контентом[33]. Кожен з цих підходів має свої особливості та переваги, які роблять їх придатними для різних типів проектів та організацій.

CMS з відкритим кодом, такі як WordPress і Drupal, забезпечують користувачам можливість безкоштовного доступу до коду системи, що дозволяє модифікувати і адаптувати платформу під індивідуальні потреби. Ці системи підтримуються великими спільнотами розробників, які надають численні розширення, плагіни та теми, спрощуючи розширення функціоналу сайту без значних інвестицій. Відкритий код також сприяє швидкому виправленню помилок та поширенню інновацій через спільноту.[4]

Комерційні CMS, такі як Sitecore та Adobe Experience Manager, пропонують більш інтегровані рішення з вищим рівнем підтримки та безпеки. Вони часто включають спеціалізовані інструменти для маркетингу, аналітики та персоналізації, які є ідеальними для великих підприємств, що мають складні вимоги до управління веб-контентом. Ці системи можуть мати вищу вартість власності, але вони забезпечують значні переваги у термінах надійності та масштабованості, а також можуть бути краще адаптовані до вимог корпоративної безпеки та відповідності стандартам.

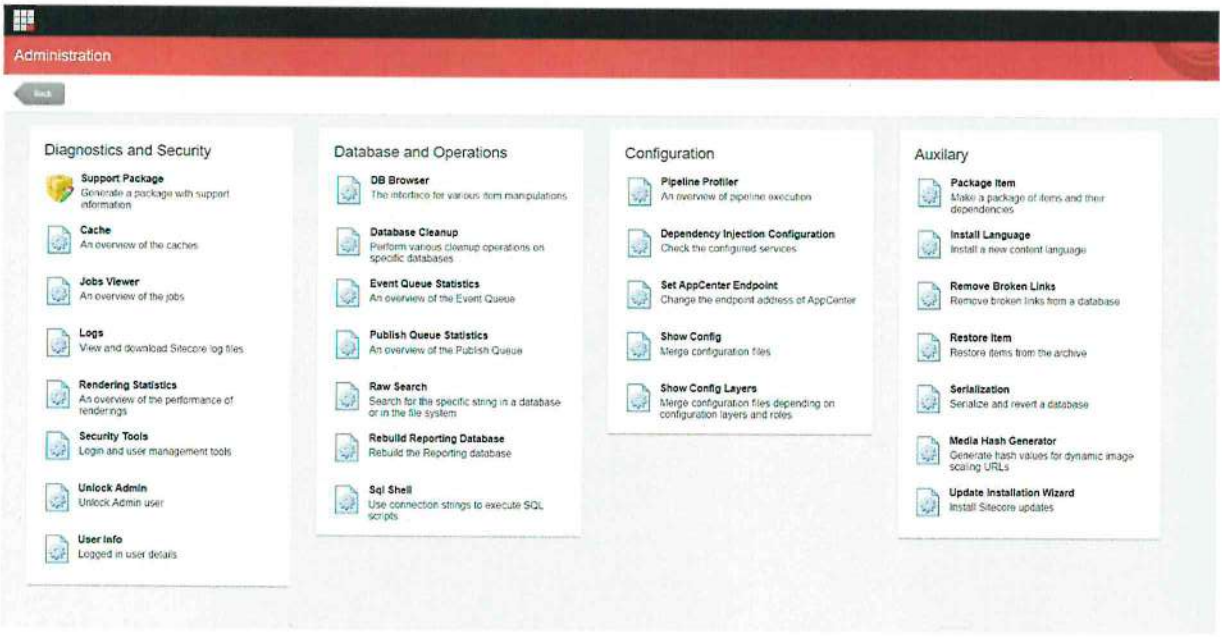


Рисунок 1.3 – Приклад комерційної CMS Sitecore

Статичні генератори сайтів та динамічні CMS

Статичні генератори сайтів, як Jekyll, працюють шляхом створення готових HTML-сторінок, які можуть бути швидко завантажені без потреби в обробці на сервері[5]. Це робить статичні сайти швидшими, безпечнішими та менш залежними від серверних ресурсів, що є великою перевагою для сайтів з високим трафіком або для тих, хто шукає простоту управління та високу надійність.

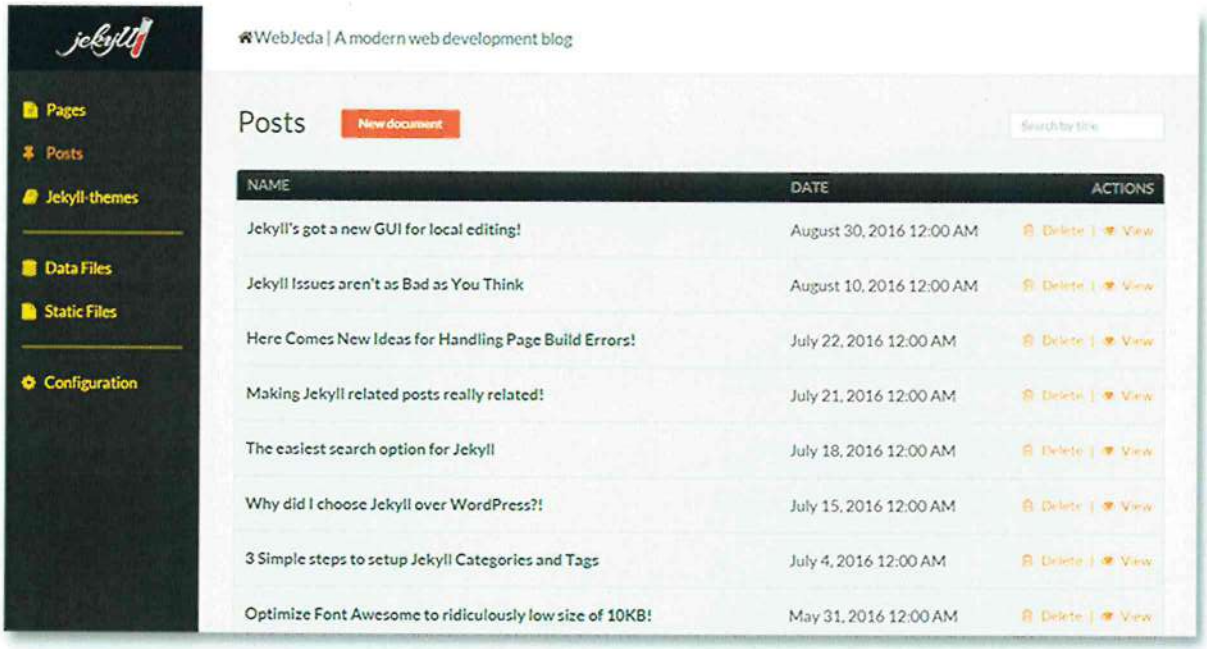


Рисунок 1.4 – Статичний генератор сайтів Jekyll

Динамічні CMS, як WordPress, використовують бази даних для зберігання контенту та динамічно генерують сторінки за запитами користувачів. Це надає значні переваги для сайтів, які потребують частого оновлення контенту, інтерактивності з користувачами або персоналізації контенту[6]. Хоча динамічні CMS можуть вимагати більшої обчислювальної потужності та ресурсів сервера, вони надзвичайно гнучкі та можуть підтримувати широкий спектр додатків і взаємодій на сайті.

Ця класифікація допомагає зрозуміти, як різні типи CMS можуть задовольнити специфічні потреби проекту, відповідаючи вимогам до функціональності, масштабованості та управління.

1.2 Технології створення веб-сайтів: HTML, CSS, JavaScript

Створення сучасних веб-сайтів базується на трьох основних технологіях: HTML, CSS і JavaScript. Кожна з цих технологій відіграє ключову роль у формуванні структури, дизайну та інтерактивності сайтів.

HTML (HyperText Markup Language) є фундаментом кожного веб-сайту. Як мова розмітки, вона дозволяє структурувати контент на веб-сторінках[7], включаючи текст, зображення, відео, та інші елементи. HTML організовує контент у вигляді блоків або елементів, кожен з яких має свої теги, атрибути та вміст.

CSS (Cascading Style Sheets) задає стиль веб-сторінок. Вона контролює візуальний вигляд сайту, включаючи шрифти, кольори, відступи та інші аспекти дизайну. Через CSS, розробники можуть маніпулювати елементами HTML для досягнення потрібного візуального стилю.

JavaScript приносить інтерактивність на веб-сайти. Ця мова програмування дозволяє створювати анімації, обробляти події користувача, валідувати форми, динамічно змінювати контент без перезавантаження сторінки через AJAX та багато іншого[8]. JavaScript взаємодіє з HTML та CSS для зміни вигляду та поведінки сторінок у реальному часі.

Розширення базових можливостей цих технологій забезпечується через використання сучасних фреймворків та бібліотек.

Фреймворки CSS такі як Bootstrap та Foundation пропонують розробникам готові компоненти та утиліти для швидкого та ефективного респонсивного дизайну. Вони дозволяють зосередитись на унікальних аспектах дизайну, мінімізуючи час, витрачений на базове стилізування.

JavaScript фреймворки як Angular, React, та Vue.js спрощують розробку інтерактивних та динамічних веб-додатків. Вони надають потужні інструменти для роботи з даними, компонентно-орієнтованого програмування та створення односторінкових додатків (SPA)[9].

jQuery продовжує бути популярним вибором для спрощення DOM маніпуляцій, обробки подій та анімації, хоча і втрачає свої позиції перед сучасними бібліотеками та фреймворками.

Препроцесори CSS як Sass та Less, забезпечують розширені можливості для ефективнішого управління стилями. За допомогою змінних, вкладених правил, міксинів, вони роблять код більш організованим та легшим для підтримки[10].

Ця комбінація технологій і інструментів формує солідну основу для розробки сучасних, швидких і візуально привабливих веб-сайтів, які відповідають вимогам користувачів і бізнесу. Різноманітність доступних фреймворків і інструментів дозволяє розробникам обирати найкращі рішення для кожного конкретного проекту, враховуючи його унікальні потреби і цілі.

1.3 Переваги використання CMS у розробці сайтів

Системи управління контентом (CMS) зіграли ключову роль у розвитку веб-дизайну та розробці, надаючи зручні та ефективні інструменти для створення та управління веб-сайтами. Основні переваги використання CMS включають наступні аспекти:

Зручність управління контентом. CMS дозволяє користувачам, які не мають технічних знань, легко створювати, редагувати та публікувати контент. Це означає, що зміст сайту може бути оновлений без необхідності втручання розробників або дизайнерів. Інтуїтивно зрозумілі інтерфейси, такі як WYSIWYG (What You See Is What You Get) редактори, дозволяють користувачам форматувати контент так, як він буде виглядати на сайті[32].

Економія часу та ресурсів. Завдяки CMS розробка та управління веб-сайтом стають значно швидшими та ефективнішими. Шаблони дизайну та

плагіни надають можливість швидко змінювати зовнішній вигляд сайту та додавати нові функції без повного перепроєктування сторінок. Це дозволяє бізнесам швидко реагувати на зміни в ринкових умовах або користувацьких перевагах.

Підвищена сумісність. Сучасні CMS зазвичай включають респонсивні теми, які автоматично адаптують веб-сайт до різних типів пристроїв, включаючи мобільні телефони та планшети. Це забезпечує кращий досвід користувача та підвищує доступність сайту.

Оптимізація для пошукових систем. Багато CMS мають вбудовані інструменти для SEO (пошукової оптимізації), які допомагають веб-сайтам краще ранжуватися в пошукових системах. Ці інструменти можуть включати можливості для оптимізації мета тегів, створення чистих URL, і вбудовану аналітику для відстеження поведінки відвідувачів.

Безпека. Постійні оновлення безпеки, які надають розробники CMS, забезпечують захист від хакерських атак і зловмисного програмного забезпечення. Крім того, системи управління контентом дозволяють налаштовувати рівні доступу для різних користувачів, що є важливим для компаній, які мають великі команди або зовнішніх контент-менеджерів.

Масштабованість. CMS зазвичай підтримують масштабування від простих блогів до великих корпоративних порталів. Це забезпечує гнучкість у виборі та використанні платформи, яка може рости разом із потребами бізнесу.

Завдяки цим перевагам, CMS є важливим інструментом у сучасній веб-розробці, дозволяючи компаніям ефективно управляти своїми онлайн-ресурсами та надавати високоякісний контент своїм користувачам.

1.4 Безпека у системах управління веб-контентом

Безпека в системах управління веб-контентом (CMS) є вирішальним фактором для забезпечення надійності та стабільності веб-сайтів. Оскільки CMS широко використовуються для різних онлайн платформ, від корпоративних сайтів до блогів, важливо забезпечити високий рівень захисту від потенційних загроз[11].

Першочерговим заходом безпеки є своєчасне оновлення CMS та її компонентів, таких як плагіни та теми. Розробники постійно виявляють та усувають вразливості, і оновлення допомагають втілити ці виправлення на вже існуючих сайтах, тим самим знижуючи ризик використання цих вразливостей зловмисниками.

Важливим аспектом є також сильні паролі та ефективне управління доступом. Використання складних паролів, поєднання цифр, літер великого та малого регістру та спеціальних символів може значно ускладнити несанкціонований доступ до адміністративної панелі CMS[12]. Керування правами доступу дозволяє обмежити можливості різних користувачів, забезпечуючи, що кожен може взаємодіяти лише з тими частинами системи, до яких має відповідні права.

На додаток, безпека CMS може бути підсилена за допомогою впровадження HTTPS, що забезпечує шифрування даних, переданих між користувачем та сервером. Це особливо критично важливо для сайтів, які обробляють конфіденційну інформацію, таку як дані кредитних карт або особисті дані. Шифрування запобігає можливості перехоплення та зчитування інформації третіми особами.

Для мінімізації впливу можливих кібератак та технічних збоїв, регулярне резервне копіювання даних є невід'ємною частиною стратегії безпеки. Резервні копії дозволяють швидко відновити втрачену інформацію та повернути сайт до нормального стану після атаки або помилки.

Крім того, обмеження можливостей завантаження файлів користувачами може запобігти використанню сайту для розповсюдження шкідливого коду. Налаштування сервера для фільтрації та обмеження типів файлів, які можна завантажувати, знижує ризик автоматичного виконання потенційно небезпечних скриптів[13].

Загалом, забезпечення безпеки систем управління веб-контентом вимагає інтегрованого підходу, який включає технічні, адміністративні та організаційні заходи. Впровадження суворих політик безпеки та їх дотримання допоможуть забезпечити надійну захист від загроз, що постійно еволюціонують у цифровому світі.

РОЗДІЛ 2. АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Аналіз ринку веб-магазинів іграшок

У сучасному цифровому світі веб-магазини іграшок іграють ключову роль у роздрібній торгівлі, надаючи батькам можливість швидко та ефективно обирати і купувати іграшки без необхідності відвідувати фізичні магазини. Системи керування веб-контентом (CMS) надають унікальні можливості для створення та адміністрування веб-магазинів, пропонуючи інструменти для гнучкого управління контентом, каталогами продуктів та інтерактивними елементами, які залучають відвідувачів[14]. В аналізі ринку веб-магазинів іграшок, реалізованих на CMS, можна виділити кілька платформ, які успішно використовують ці можливості.

Shopify є однією з найпопулярніших CMS для електронної комерції, що дозволяє користувачам швидко запускати та управляти онлайн-магазинами. Багато веб-магазинів іграшок на Shopify характеризуються високою налаштовуваністю, інтуїтивно зрозумілими інтерфейсами та зручністю управління продуктами. Shopify також пропонує різноманітні SEO інструменти, що допомагають магазинам підвищувати свою видимість в пошукових системах.

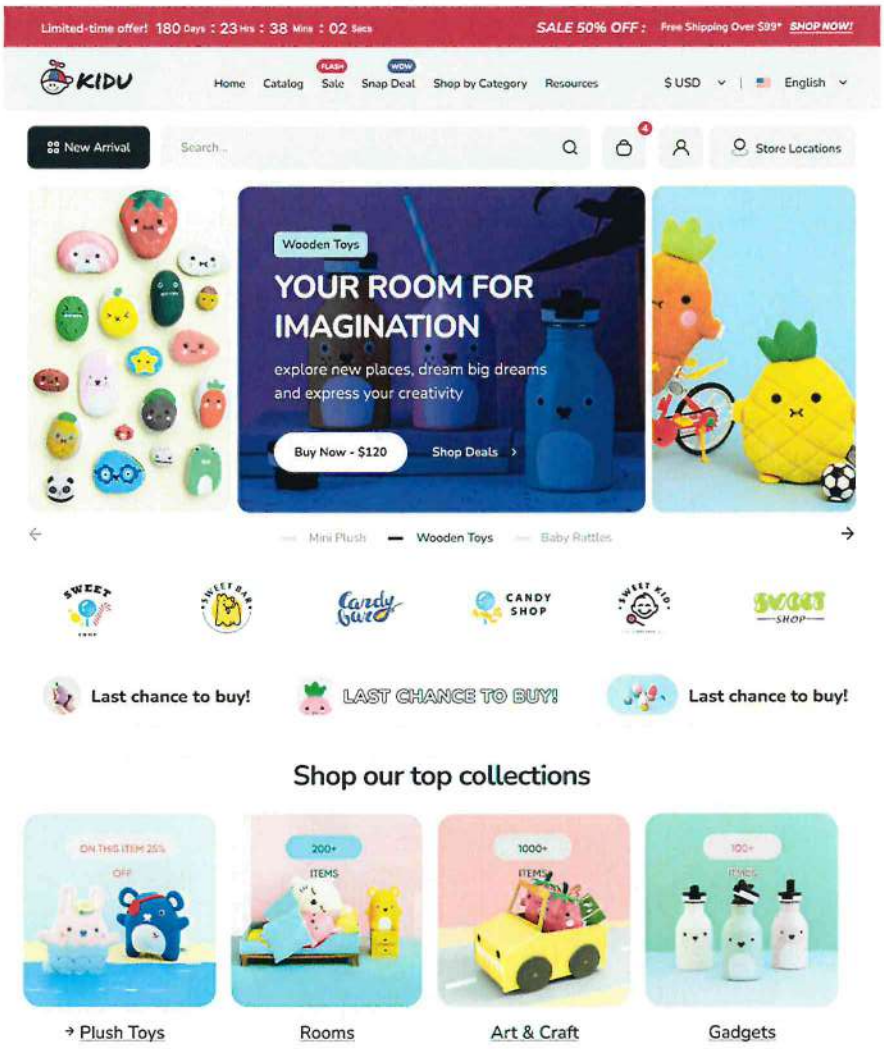


Рисунок 2.1 – Приклад створеного магазину іграшок з використанням Shopify

Magento є ще однією платформою, яка забезпечує розширені можливості для створення веб-магазинів іграшок. Завдяки гнучкій архітектурі та могутнім інструментам для розробників, Magento дозволяє створювати масштабовані та повністю налаштовані рішення, які можуть задовольнити потреби великих ритейлерів[15]. На Magento можна зустріти багато

багатомовних сайтів з розширеною системою управління замовленнями та клієнтською базою.

WooCommerce, як доповнення до WordPress, також широко використовується для створення веб-магазинів іграшок. Ця платформа відома своєю простотою інтеграції, великою кількістю додаткових плагінів та тем, які дозволяють легко адаптувати сайт під індивідуальні потреби магазину. WooCommerce вважається одним з найкращих варіантів для малого та середнього бізнесу через низькі витрати на старт ті відносно просте управління сайтом. WooCommerce надає велику гнучкість у налаштуванні продуктів, що дозволяє детально описувати кожну іграшку, включаючи варіанти, різноманітні атрибути та спеціальні акційні пропозиції, що є важливим для залучення та утримання клієнтів.

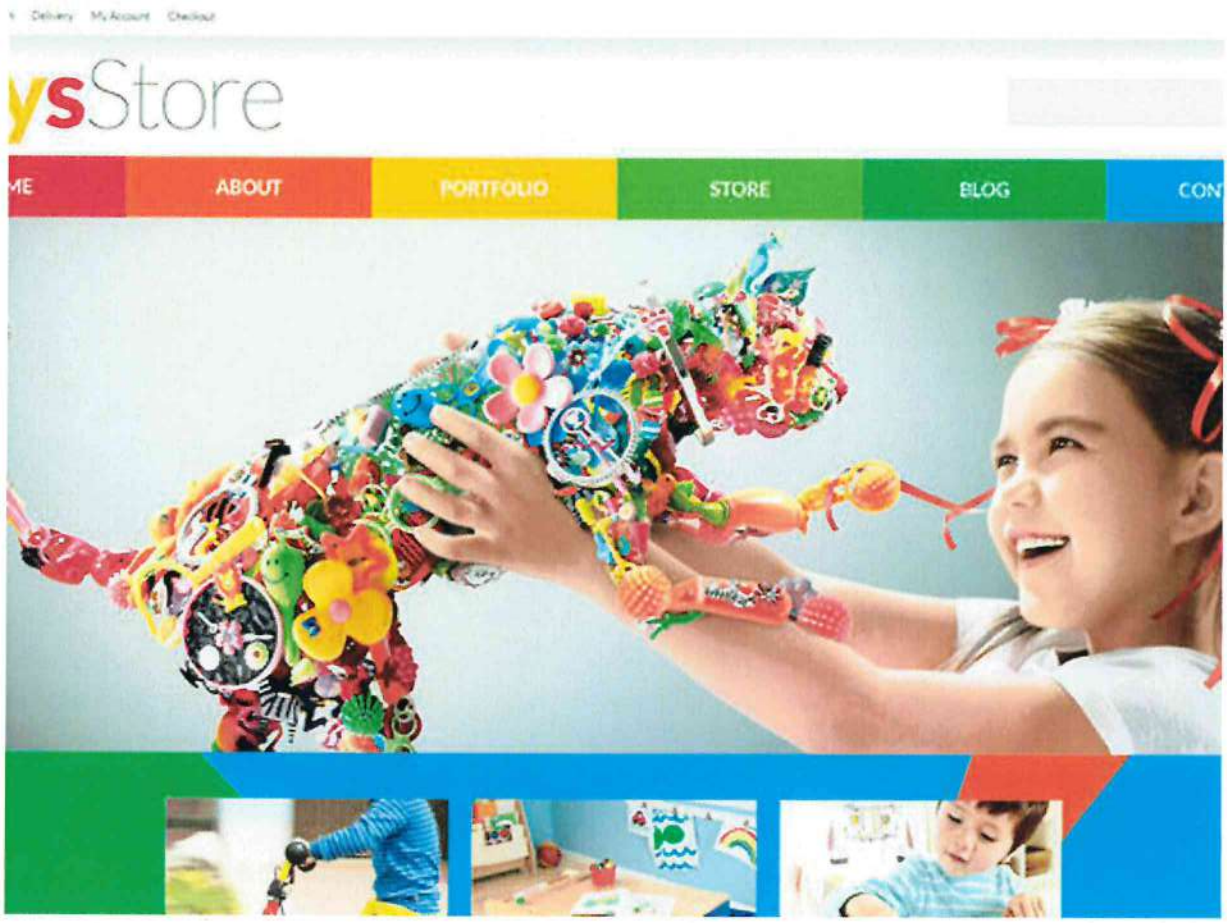


Рисунок 2.2 – Приклад створеного магазину іграшок з використанням WooCommerce

Не можна не згадати про **Contentful**, який є headless CMS і надає велику свободу для розробників і маркетологів у створенні кастомізованого контенту без обмежень традиційної CMS. Contentful відокремлює вміст від представлення, що дозволяє управляти контентом і розповсюджувати його через різні канали[16]. Це особливо корисно для брендів, які хочуть надати унікальний досвід покупок через різні платформи, включаючи мобільні додатки та соціальні мережі.

На основі проведеного аналізу веб-магазинів іграшок, які реалізовані на різних CMS платформах, можна виявити ключові сильні та слабкі сторони кожної з них. Для наочності складемо таблицю порівняння, яка дозволить детальніше розглянути особливості кожного магазину на прикладах Shopify, Magento, WooCommerce та Contentful.

Таблиця 2.1

Порівняння веб-магазинів іграшок на CMS платформах

Параметр	Shopify	Magento	WooCommerce	Contentful
Асортимент	Широкий вибір, легко масштабується	Великий вибір, гнучке управління каталогом	Широкий асортимент, легке додавання продуктів	Гнучке управління контентом, інтеграція з додатками
Цінова політика	Конкурентні ціни, різноманітні тарифні плани	Висока налаштованість, вимагає більше інвестицій	Економічно вигідно для МСБ	Залежить від обраної підписки
Логістика	Інтеграція з численними логістичними сервісами	Гнучкі налаштування доставки	Проста інтеграція з основними службами доставки	Не зосереджена на логістиці, потребує додаткової інтеграції
Технології	Вбудовані SEO інструменти,	Високий рівень кастомізації,	Легкість використання, велика кількість плагінів	API-центрична,

	розширені аналітичні функції	підтримка комплексних функцій		підтримка омніканальності
Користувацький досвід	Інтуїтивно зрозумілий інтерфейс	Потребує технічних знань для налаштування	Користувацький досвід залежить від вибраних тем і плагінів	Висока адаптивність контенту, ідеально для персоналізації

Аналіз таблиці порівняння

Shopify пропонує багатофункціональне рішення для електронної комерції, що ідеально підходить для бізнесів будь-якого розміру завдяки своїй масштабованості та легкості використання. Його інтеграція з різноманітними платформами логістики та маркетингу робить Shopify особливо привабливим для швидкого запуску продуктів[34].

Magento вирізняється гнучкістю у налаштуванні та управлінні складними каталогами продукції. Ця CMS підходить для великих рітейлерів, які потребують високого рівня кастомізації і мають доступ до ресурсів для розвитку та підтримки своїх сайтів.

WooCommerce є відмінним вибором для малих та середніх підприємств. Ця платформа пропонує низьку вартість впровадження та велику кількість доступних розширень, що дозволяє легко адаптувати сайт до потреб бізнесу без великих вкладень.

Contentful пропонує унікальний підхід зі своєю headless архітектурою, що дозволяє відокремити контент від дизайну та розповсюджувати його через різні канали. Це ідеальне рішення для тих, хто шукає гнучкість у

мультіканальному маркетингу та хоче створювати персоналізований користувацький досвід.

Цей аналіз підкреслює важливість вибору CMS з огляду на специфічні потреби бізнесу та дозволяє зрозуміти, які платформи найкраще відповідають різним вимогам в контексті онлайн-продажу іграшок.

2.2 Функціональні можливості існуючих CMS

Розглядаючи функціональні можливості сучасних CMS, які використовуються для розробки веб-магазинів іграшок, важливо зосередитись на технічних деталях, які визначають здатність кожної системи задовольнити специфічні потреби бізнесу. Вибір CMS має велике значення, оскільки він впливає на швидкість розробки, масштабування, інтеграцію з іншими сервісами, а також на загальну безпеку та легкість управління веб-магазином.

Shopify – це хмарна платформа, яка забезпечує широкий набір функціональних можливостей за допомогою плагінів та тем[18]. Ця CMS має вбудовану підтримку різних методів оплати, що робить її вкрай привабливою для швидкого старту нових магазинів. Shopify також автоматично оновлюється, забезпечуючи високий рівень безпеки та новітні функції для користувачів без потреби їх ручного оновлення.

Magento відома своєю гнучкістю і потужністю, пропонуючи розширені можливості для кастомізації. Ця платформа підтримує складну логістику та управління декількома магазинами і складами з одного інтерфейсу. Magento є відмінним вибором для великих електронних торговців, які потребують детального контролю над кожним аспектом своєї онлайн-присутності. Однак, для повного використання потенціалу Magento необхідні серйозні технічні знання та ресурси.

WooCommerce є додатком для WordPress, що дозволяє користувачам легко перетворити будь-який сайт на повнофункціональний веб-магазин. Головною перевагою WooCommerce є його тісна інтеграція з WordPress, що дозволяє використовувати широкий спектр плагінів для розширення функціональності сайту. Ця CMS підтримує необмежену кількість продуктів і користувачів, інтеграцію з багатьма платіжними системами, і налаштування SEO.

Contentful як headless CMS пропонує унікальний підхід до управління контентом. Відокремлюючи "голову" (фронтенд) від "тіла" (бекенд), Contentful дає розробникам свободу вибору технологій для фронтенду, покращуючи таким чином швидкість завантаження сайтів та інтеграцію з різними системами і додатками[19]. Ця платформа ідеально підходить для проектів, які потребують великої гнучкості у публікації контенту через різні канали, включаючи мобільні додатки, інтерактивні дисплеї, і навіть IoT пристрої.

Обравши відповідну CMS, можна не тільки оптимізувати процеси управління інтернет-магазином, але й значно підвищити якість користувацького досвіду, забезпечивши високу швидкість роботи сайту, легкість навігації та безпеку транзакцій. Вибір правильної CMS дозволяє компаніям бути гнучкими у відповіді на змінні вимоги ринку та потреби клієнтів.

2.3 Критерії вибору CMS для веб-магазину іграшок

При виборі системи керування контентом (CMS) для веб-магазину іграшок важливо звернути увагу на ряд критеріїв, які допоможуть забезпечити ефективність, безпеку та зручність використання платформи. Основні аспекти, які слід розглянути, включають інтуїтивність інтерфейсу, масштабованість,

безпеку, підтримку мультимедіа, інтеграцію SEO, технічну підтримку та вартість впровадження та експлуатації CMS[20].

Перш за все, інтуїтивно зрозумілий інтерфейс користувача є критично важливим для ефективного управління веб-магазином. Адміністратори магазину повинні мати можливість легко додавати, редагувати та видаляти продукти, управляти запасами та обробляти замовлення без необхідності глибоких технічних знань.

Масштабованість також є важливим аспектом, оскільки веб-магазин має потенціал до росту. Вибрана CMS має забезпечити легке масштабування від невеликого кількості продуктів до тисячі товарних позицій і обробки великої кількості транзакцій без втрати продуктивності.

Безпека є ще одним вирішальним критерієм, оскільки веб-магазини збирають та обробляють великі обсяги конфіденційних даних користувачів та платіжної інформації. Наявність сильних засобів захисту даних, регулярні оновлення безпеки та відповідність стандартам PCI DSS є обов'язковими для захисту цієї інформації.

Підтримка мультимедіа важлива для веб-магазину іграшок, оскільки якісні зображення та відео значно покращують візуальну презентацію товарів. CMS має надавати легкі інструменти для інтеграції та оптимізації мультимедійного контенту, що допоможе підвищити залученість покупців.

SEO оптимізація є невід'ємною частиною успішної онлайн-присутності, тому вибрана CMS має містити вбудовані інструменти для управління SEO, такі як налаштування метатегів, карти сайту, та URL, оптимізованих для пошукових систем[21].

Врешті-решт, при виборі CMS для веб-магазину іграшок також слід враховувати вартість впровадження та подальшої підтримки системи, а також

якість технічної підтримки, яку надає постачальник. Вибір правильної CMS впливатиме на довгострокову ефективність і успіх онлайн-бізнесу.

2.4 Аналіз користувацького інтерфейсу та досвіду взаємодії на існуючих сайтах

Для аналізу користувацького інтерфейсу та досвіду взаємодії на існуючих сайтах веб-магазинів іграшок варто звернути увагу на декілька важливих аспектів, які значно впливають на задоволеність користувачів та ефективність використання ресурсів. Ось основні критерії, які слід розглянути:

1. **Навігація та доступність** - Чи легко користувачам орієнтуватися на сайті, чи інтуїтивно зрозуміле розміщення меню та інших елементів управління? Наявність чіткої, логічно структурованої навігації допомагає користувачам швидко знаходити потрібні товари, що важливо для підтримки високого рівня залученості та зниження відсотка відмов[23].
2. **Консистентність дизайну** - Чи зберігається єдність стилів, шрифтів, кольорів та макетів на всіх сторінках сайту? Консистентність дизайну не тільки сприяє кращому сприйняттю бренду, але й підвищує впізнаваність елементів управління, що робить взаємодію з сайтом більш комфортною та передбачуваною.
3. **Швидкість завантаження** - Як швидко завантажуються сторінки сайту? Висока швидкість завантаження є критично важливою для утримання користувачів, особливо коли вони використовують мобільні пристрої. Оптимізація зображень, скорочення JavaScript та CSS, використання кешування можуть значно покращити цей показник.
4. **Адаптивність дизайну** - Чи коректно відображається сайт на різних пристроях, включаючи настільні комп'ютери, планшети та смартфони?

Адаптивний дизайн забезпечує зручне і приємне використання ресурсу без залежності від розміру екрану, що є ключовим для залучення та утримання користувачів.

5. **Інтерактивність** - Як сайт взаємодіє з користувачем? Чи є інтерактивні елементи, такі як відеоогляди продуктів, 3D-моделі, віртуальні примірки, або інтерактивні ігри, які можуть підвищити залученість та сприяти кращому розумінню продуктів? Інтерактивні елементи можуть значно покращити досвід користувачів, стимулюючи їхнє бажання здійснювати покупки.

Аналізуючи ці аспекти, можна отримати глибоке розуміння якості користувацького інтерфейсу та досвіду взаємодії на існуючих сайтах[24]. Це дозволить виявити потенційні точки зростання та аспекти, які потребують удосконалення, що в кінцевому підсумку призведе до підвищення задоволеності клієнтів та збільшення продажів.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ

3.1 Проектування архітектури сайту

Для створення ефективного веб-магазину іграшок важливо уважно продумати архітектуру сайту, використовуючи сучасні технології та фреймворки, що забезпечують швидкість, безпеку та зручність для користувачів. Розглянемо проектування архітектури системи на прикладі використання таких технологій, як React, Nest.js, Node.js, TypeScript, а також інтеграції з Contentful через RESTful API[25].

Фронтенд-архітектура

Фронтенд-архітектура веб-сайту відіграє ключову роль у забезпеченні відмінного користувацького досвіду, швидкості завантаження та зручності інтеракції з сайтом. У цьому проекті веб-магазину іграшок основою для розробки клієнтської частини вибрані React і Next.js. React є однією з найпопулярніших бібліотек JavaScript для створення інтерактивних користувацьких інтерфейсів, яка дозволяє розробникам створювати ефективні, взаємопов'язані компоненти, що реагують на дії користувачів без потреби перезавантаження сторінки. Це сприяє створенню плавних та швидких застосунків, підвищуючи загальну задоволеність користувачів.

Next.js, у свою чергу, доповнює можливості React, пропонуючи такі функції, як серверне рендерення (SSR), що дозволяє забезпечити більш швидке завантаження початкового контенту і покращує SEO, оскільки контент генерується на сервері і відразу доступний для індексації пошуковими

системами. Це робить Next.js ідеальним вибором для проектів, де швидкість завантаження і оптимізація для пошукових систем є пріоритетними.

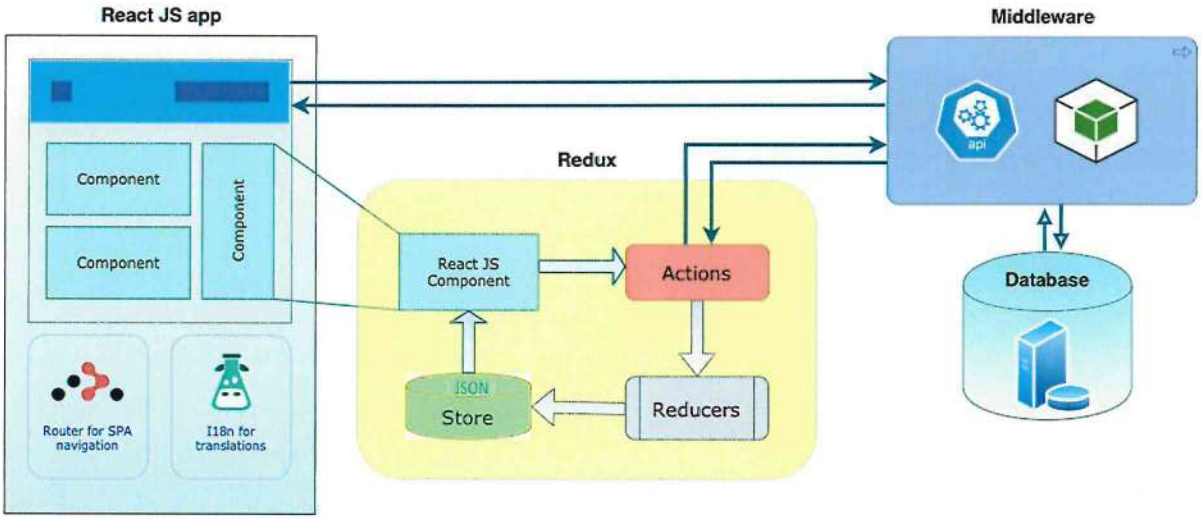


Рисунок 3.1 – Архітектура React

Додавання TypeScript до стеку технологій фронтенду вносить строгу типізацію в JavaScript-код, що сприяє зниженню помилок на етапі компіляції, полегшує супровід коду та підвищує його якість. TypeScript визначає можливі помилки ще до виконання скриптів у браузері, дозволяючи розробникам більш впевнено управляти даними та взаємодіями великих і складних додатків.

Для стилізації інтерфейсу використовується TailwindCSS — утилітарний CSS-фреймворк, який надає розробникам величезний набір утиліт для швидкого створення адаптивних дизайнів без необхідності писати багато власного CSS[26]. Це значно прискорює процес розробки, дозволяє легко модифікувати вигляд компонентів і забезпечує високий рівень консистентності та адаптивності дизайну на різних пристроях і роздільних здатностях.

Таким чином, використання цього набору технологій та інструментів для фронтенд-архітектури не лише забезпечує ефективне відображення контенту

та інтерфейсу, але й відкриває широкі можливості для оптимізації продуктивності, доступності та користувацького досвіду в майбутньому.

Бекенд-архітектура

Бекенд-архітектура сайту заснована на використанні Nest.js і Node.js, що дозволяє створити ефективний і гнучкий серверний додаток[27].

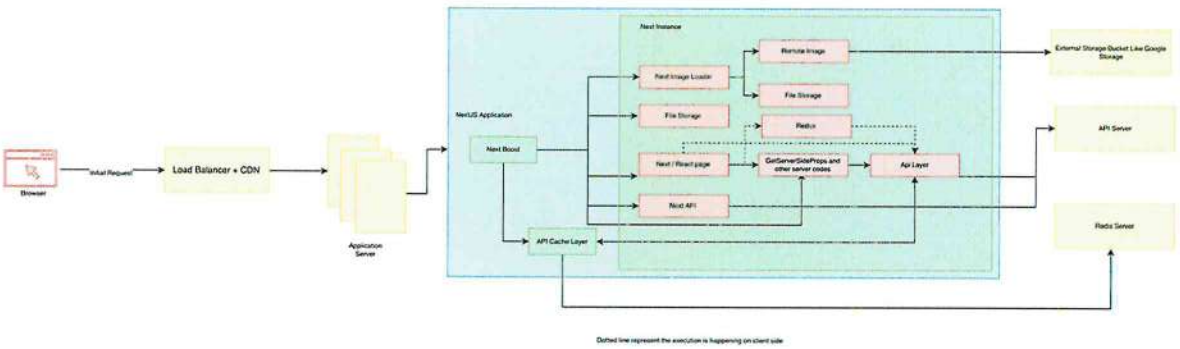


Рисунок 3.2 – Архітектура Nest.js

Нижче наведено основні аспекти архітектури, які були інтегровані:

1. **Інтегрована підтримка REST API та GraphQL** - Nest.js дозволяє легко імплементувати REST API для обробки HTTP запитів. Це надзвичайно важливо для оперативної взаємодії між клієнтською та серверною частинами. Крім того, фреймворк підтримує GraphQL, що забезпечує більш динамічний та ефективний підхід до управління даними, дозволяючи клієнтам точно специфікувати, які дані їм потрібні.
2. **Підтримка мікросервісної архітектури** - Nest.js відмінно підходить для розробки за мікросервісним підходом, що дозволяє розподіляти

обробку даних та бізнес-логіку по окремим сервісам. Це сприяє кращій масштабованості та легшому управлінню додатком, особливо коли система розростається.

3. **Застосування TypeScript для підвищення безпеки та стабільності коду** - Використання TypeScript у Nest.js дозволяє розробникам використовувати переваги строгої типізації для підвищення якості коду та зменшення кількості помилок на етапі розробки[28]. Строга типізація також полегшує рефакторинг і масштабування додатку.

Ці елементи бекенд-архітектури забезпечують міцну основу для створення надійного та продуктивного веб-магазину іграшок. Використання Nest.js і Node.js у поєднанні з іншими сучасними технологіями створює ефективне середовище для розробки, яке може легко адаптуватися до змінних вимог ринку та користувацьких вподобань.

Інтеграція з Contentful через RESTful API

Інтеграція з Contentful, який є headless CMS, забезпечує гнучке управління контентом веб-магазину. Використання RESTful API дозволяє легко отримувати, змінювати та синхронізувати дані про продукти, категорії, блоги та інші типи контенту з сервером[29]. Така архітектура допомагає відділяти контент від презентації, забезпечуючи більшу гнучкість у розробці та можливість використання контенту на різних платформах.

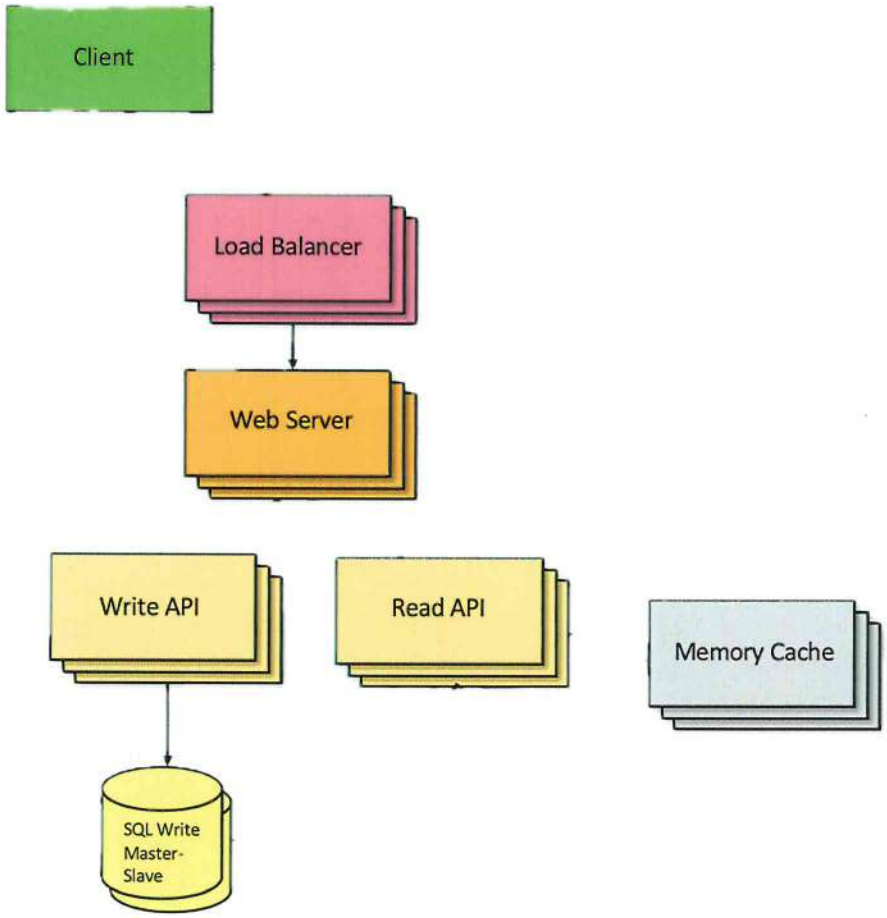


Рисунок 3.3 – RESTful API

Ці технології та підходи до проектування архітектури сайту дозволяють створити масштабований, ефективний та безпечний веб-магазин іграшок, який забезпечує відмінний користувацький досвід і високу швидкість роботи на всіх типах пристроїв[31]. Інтеграція з сучасними веб-технологіями та гнучкими інструментами управління контентом гарантує, що веб-магазин буде легко адаптуватися до змінюваних вимог ринку і користувацьких переваг.

3.2 Розробка користувацького інтерфейсу

Проект веб-магазину іграшок організовано з чітко визначеною структурою, що допомагає в ефективному управлінні та розширенні проекту. Структура проекту поділяється на кілька ключових директорій, кожна з яких відіграє важливу роль в розробці веб-сайту.

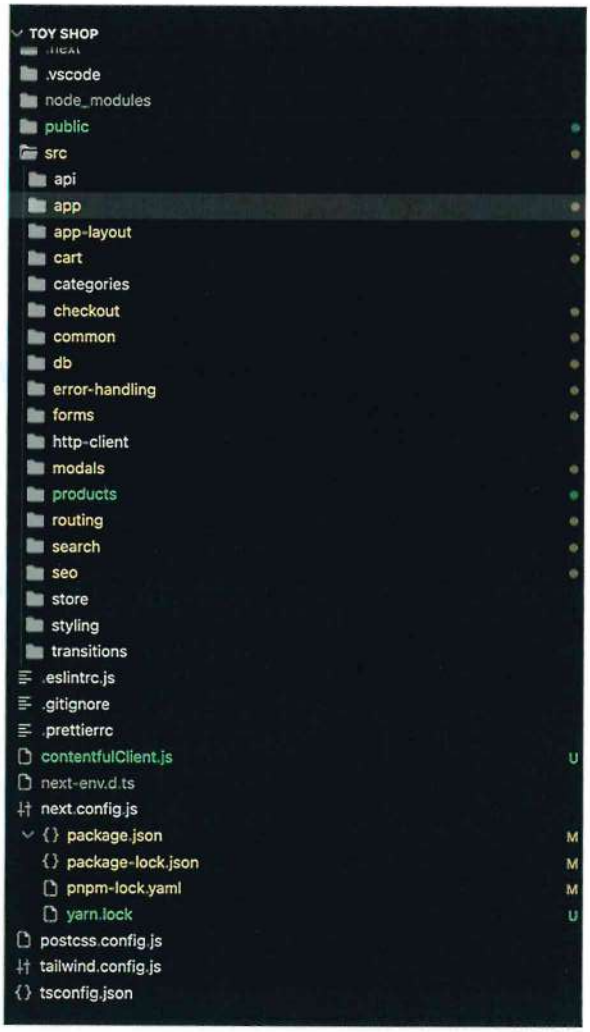


Рисунок 3.4 – Структура розробленного проекту

Каталог **src** є головним контейнером для усіх розроблюваних компонентів і модулів. Він включає підкаталоги, такі як **api**, де розміщені файли, що відповідають за взаємодію з серверною частиною через API. Це

дозволяє централізовано керувати всіма запитами та відповідями, що взаємодіють з бекендом, забезпечуючи обробку даних, які вносяться чи запитуються користувачами.

Каталог **app** містить основну логіку користувацьких інтерфейсів та взаємодію з користувачем, включаючи різноманітні компоненти React, які використовуються для побудови сторінок веб-сайту. Компоненти поділені за функціональністю, що дозволяє забезпечити легку скалабельність та перевикористання коду.

Також проект містить каталоги **common** та **components**, які включають загальні утиліти та відновлювані UI компоненти відповідно. Це сприяє стандартизації та оптимізації розробки, забезпечуючи легке управління стилями та поведінкою компонентів на всьому сайті.

Routing та **services** каталоги відповідають за управління маршрутизацією та бізнес-логікою додатку. Це включає налаштування шляхів, які користувачі можуть слідувати, та визначення логіки обробки даних, що необхідна для реалізації функціоналу веб-магазину.

Останній каталог **store** управляє станами компонентів за допомогою Redux, забезпечуючи єдине джерело правди для всієї інформації, яка потребує глобального доступу і реактивної взаємодії у межах проекту.

Ця структура проекту не тільки дозволяє ефективно розподіляти завдання між різними аспектами проекту, але й забезпечує масштабування та адаптацію проекту до змінних вимог користувачів та технологічних умов ринку.

Ключові компоненти системи

Для веб-магазину іграшок розроблено низку компонентів, кожен з яких відіграє важливу роль у забезпеченні функціональності та користувацького досвіду. Нижче описано детально основні компоненти інтерфейсу користувача, їх роль і як вони взаємодіють між собою.

Компонент AppLayoutRoot та Hero

В рамках розробки веб-магазину іграшок велику увагу приділено створенню ефективного та зручного користувацького інтерфейсу, який був реалізований за допомогою низки ключових компонентів. Центральним елементом архітектури є компонент **AppLayoutRoot**, який виконує функцію основної обгортки для всіх сторінок магазину. Цей компонент забезпечує єдину структурну базу для фронтенду, гарантуючи однорідність макету сторінок, що включає заголовки, основний вміст, і підвал. Використання CSS Grid дозволяє гнучко та ефективно структурувати вміст сторінки, розділяючи її на три основні зони: заголовок, основний вміст, та підвал, забезпечуючи тим самим чітку візуальну композицію.

Рисунок 3.5 – Реалізація компоненту Layout

Для привернення уваги користувачів на головній сторінці використовується компонент **Hero**. Цей елемент веб-дизайну є критично важливим, оскільки він є першим, на що звертає увагу користувач при вході на сайт. **Hero** може містити яскраві зображення, слайдери або відео, які демонструють актуальні акції, популярні товари або особливості товарів, що продаються. Ефектний візуальний вступ сприяє залученню відвідувачів та стимулює їх на подальше дослідження асортименту магазину.

Комбінація цих двох компонентів створює міцну основу для користувацького досвіду, спрямованого на забезпечення високої зручності та привабливості веб-магазину. **AppLayoutRoot** забезпечує структурну цілісність і однорідність дизайну сторінок, тоді як **Hero** вносить динаміку та емоційну привабливість в перше враження користувачів. Такий підхід дозволяє не тільки

втримати увагу користувача, але й спонукати його до подальшого ознайомлення з продуктовою пропозицією сайту.

Компонент Categories

Компонент **Categories** у веб-магазині іграшок служить однією з ключових функціональних частин, забезпечуючи користувачам зручний інструмент для швидкого доступу до різноманітних категорій товарів. Розміщений на головній сторінці та інших важливих секціях сайту, цей компонент значно поліпшує користувацький досвід завдяки своїй інтуїтивно зрозумілій структурі та візуальній привабливості. Кожна категорія товарів представлена у вигляді іконки або зображення з коротким описом, що допомагає користувачам легко ідентифікувати бажаний розділ товарів.

```
src > categories > <> CategoryLink.tsx > ...
      Kapood, 14 months ago | 1 author (Kapood)
1     import NextLink, { NextLinkProps } from '@/routing/NextLink';
2     import BaseImage from '@/common/BaseImage';
3
4     type CategoryLinkProps = Pick<NextLinkProps, 'href'> & {
5       imageSrc: string;
6       title: string;
7     };
8
9     export default function CategoryLink({
10      href,
11      imageSrc,
12      title,
13    }: CategoryLinkProps) {
14      return (
15        <NextLink
16          className="relative block w-full h-80 group rounded-md overflow-hidden"
17          href={href}
18        >
19          <BaseImage
20            src={imageSrc}
21            alt={title}
22            className="object-cover transition duration-700 transform group-hover:scale-110"
23            fill
24            priority
25          />
26          <div className="absolute inset-0 grid place-items-center">
27            <div className="p-6 bg-black bg-opacity-50 rounded-md">
28              <h2 className="text-white text-3xl md:text-4xl font-bold border-b-4 mb-2 text-center">
29                {title}
30              </h2>
31            </div>
32          </div>
33        </NextLink>
34      );
35    }
36
```

Рисунок 3.6 – Реалізація Categories

Динамічність компонента **Categories** полягає у його здатності адаптуватися та відображати актуальні категорії, що можуть бути отримані з сервера або визначені статично. Така гнучкість є надзвичайно важливою для підтримки актуальності асортименту та реагування на зміни у попиті споживачів без потреби в ручному оновленні даних. Візуальні елементи кожної категорії не лише забезпечують зрозумілість і доступність інформації, але й сприяють створенню естетично привабливого дизайну, що є критично важливим для залучення та утримання уваги користувачів.

Загалом, компонент **Categories** є вирішальним у створенні ефективного, зручного та приємного досвіду шопінгу в онлайн-магазині іграшок, допомагаючи користувачам швидко орієнтуватися в широкому асортименті товарів і сприяючи кращому розумінню того, що пропонує магазин.

Компонент Cart

Модуль корзини у веб-магазині іграшок включає декілька компонентів, які разом забезпечують комплексне управління покупками користувачів, роблячи процес вибору, редагування та оформлення замовлення інтуїтивно зрозумілим і зручним. Основа модуля корзини—це Redux Toolkit, який використовується для управління станом корзини через **CartSlice**, дозволяючи додавати, видаляти та змінювати кількість товарів. Цей підхід дозволяє корзині динамічно реагувати на взаємодії користувачів, оновлюючи вміст корзини в реальному часі без перезавантаження сторінки.

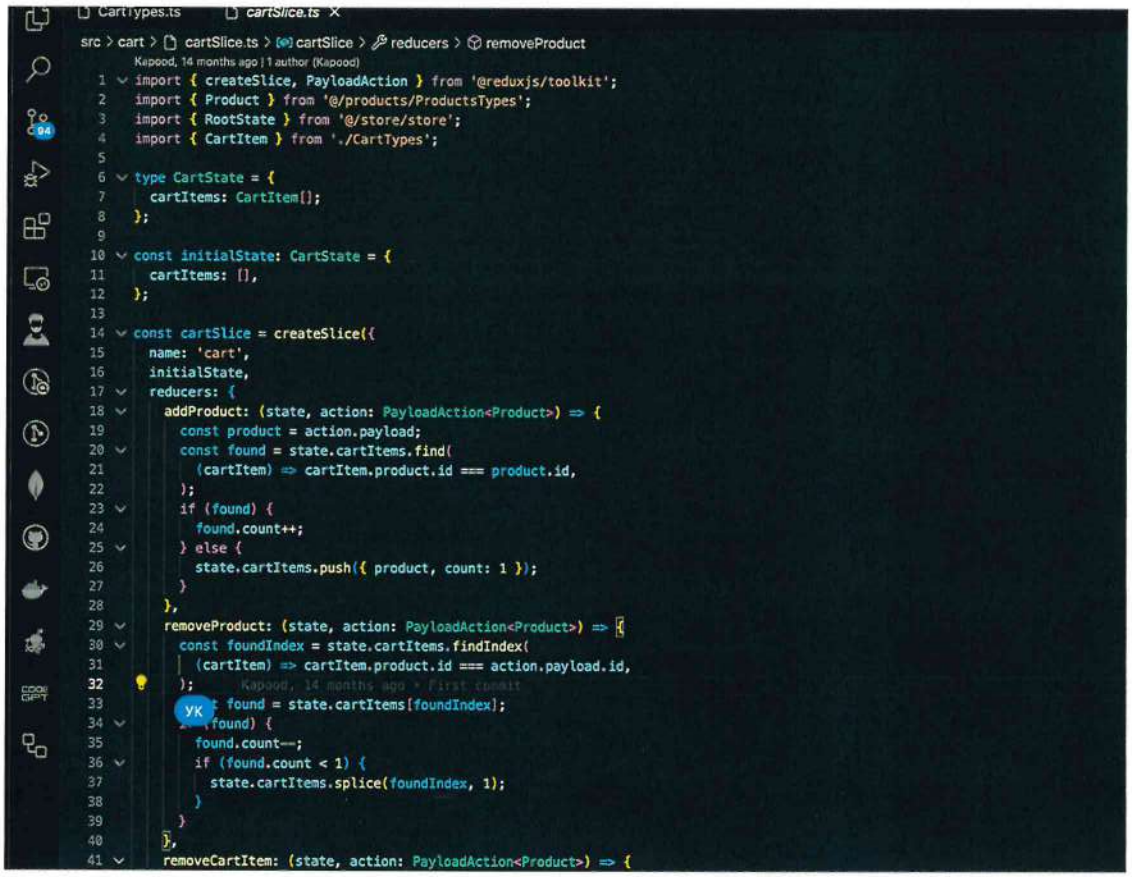


Рисунок 3.7 – Реалізация модулю

Візуальне представлення корзини реалізоване через компоненти **CartDrawer** та **CartItemList**, які відображають список товарів, що користувач додав до корзини. **CartDrawer** надає можливість користувачам доступу до деталей корзини з будь-якої сторінки сайту, підтримуючи гнучкість взаємодії та поліпшуючи користувацький досвід. Компонент **CartItemActionButtons** дозволяє користувачам змінювати кількість конкретного товару або видаляти його з корзини, що робить управління покупками простим та ефективним.

Завершення покупки спрощено завдяки компоненту **CartTotalPrice**, який підсумовує вартість всіх товарів у корзині, і **Checkout** кнопці в **CartDrawer**, яка веде користувача до оформлення замовлення. Це дозволяє користувачам швидко перейти від вибору товарів до їхнього придбання, забезпечуючи безперервний та зручний перехід від шопінгу до оформлення

замовлення. Такий інтегрований підхід в управлінні корзиною покращує загальне задоволення користувачів та сприяє збільшенню конверсії в продажах.

Компонент Checkout

В рамках розробки модулю оформлення замовлення (checkout) для веб-магазину іграшок, було створено низку компонентів, що забезпечують зручне та безпечне внесення платіжної інформації та підтвердження покупки користувачем. Основними елементами цього модулю є форми для вводу даних карти, валідації та обробки замовлення.

Компонент **CheckoutForm** включає у себе поля для введення імені та прізвища, номера кредитної картки, терміну дії та CVC-коду. Використання бібліотеки **react-hook-form** з валідатором **zod** забезпечує не тільки легкість внесення даних, але й їхню перевірку на відповідність встановленим критеріям, що сприяє безпеку транзакцій. Компонент **NumberInput** використовується для форматування номера карти та терміну придатності, дозволяючи користувачам вводити дані у відповідному форматі без помилок.

Рисунок 3.8 – Реалізація checkout

Значущу роль відіграє компонент **CardExpiryInput**, який відповідає за введення та форматування дати закінчення терміну дії карти. Він автоматично коригує введені користувачем дані, забезпечуючи введення дійсного терміну. Це знижує ризик помилок під час транзакцій і покращує користувацький досвід.

Для завершення процесу оформлення замовлення створений компонент **SubmitButton**, який взаємодіє з сервером через асинхронні запити, відправляючи усі зібрані дані на обробку та підтвердження покупки. У випадку успішного оформлення замовлення, користувачі спрямовуються на сторінку з

підтвердженням **CheckoutSuccessMessage**, яка інформує їх про успішне прийняття замовлення та надає можливість повернутися до магазину.

Компоненти common

У проекті використовується набір загальних компонентів для реалізації користувацького інтерфейсу, який включає такі елементи як **Backdrop**, **Badge**, **BaseImage**, **Button**, **Chip** і багато інших. Кожен з цих компонентів відіграє ключову роль у створенні зручної та привабливої взаємодії з користувачем.

Компонент **Backdrop** використовується для створення модальних вікон або контекстних меню, що накладаються на основний контент, тимчасово блокуючи взаємодію з іншими елементами інтерфейсу. Це дозволяє зосередити увагу користувача на конкретній дії або інформації. Він має властивість **closeOnRouteChange**, яка автоматично закриває компонент при зміні маршруту, забезпечуючи відповідність користувацького інтерфейсу очікуванням користувачів під час навігації.

Компонент **Badge**, який використовується для відображення інформації про кількість предметів або сповіщень, часто поміщається на іконках або кнопках для індикації нових повідомлень або непереглянутих елементів, як це часто робиться у кошику покупок або в модулях сповіщень.

Компонент **BaseImage** є обгорткою для елемента зображення, оптимізованим для використання у Next.js, що забезпечує ефективну обробку та відображення зображень на веб-сайті. Цей компонент використовується для забезпечення різних аспектів зображень, таких як адаптивність та відповідність сучасним веб-стандартам.

Chip – це невеликий компонент, який може відображати теги, категорії або вибори, часто з варіантами для видалення або взаємодії. Це дозволяє

користувачам швидко визначати контекстуальну інформацію або управляти елементами, які використовуються для фільтрації або сортування контенту.

Компонент ProductCard

Компонент **ProductCard** є ключовим для відображення інформації про продукти у веб-магазині іграшок. Цей компонент ефективно використовується для презентації окремих товарів на різних сторінках сайту, забезпечуючи користувачам зручний перегляд інформації та легкий доступ до детальних сторінок товарів.

ProductCard призначений для структурування візуального представлення товару, використовуючи комбінацію тексту та зображень. Основною функцією цього компонента є забезпечення відображення картинки товару, назви, та ціни, які є важливими для швидкого ознайомлення з товаром.

Зображення продукту реалізовано за допомогою компонента **BaseImage**, який інтегровано з параметрами адаптивності. Це дозволяє зображенням товарів відображатися ефективно на різних пристроях, підлаштовуючись під різні розміри екранів. Атрибути, які задані через **imageProps.responsive**, гарантують, що зображення будуть масштабуватися та підлаштовуватися під контейнер без втрати якості візуального вмісту.

Ціну товару представлено через компонент **Price**, який форматує вартість товару у зручний для сприйняття формат, підкреслюючи цінову привабливість товару.

Інтеграція з роутингом здійснюється через компонент **NextLink**, який дозволяє легко навігувати до детальної сторінки товару, використовуючи параметри товару для динамічного формування URL-адрес. Це забезпечує

зручність навігації в межах сайту та підвищує загальну інтерактивність користувацького досвіду.

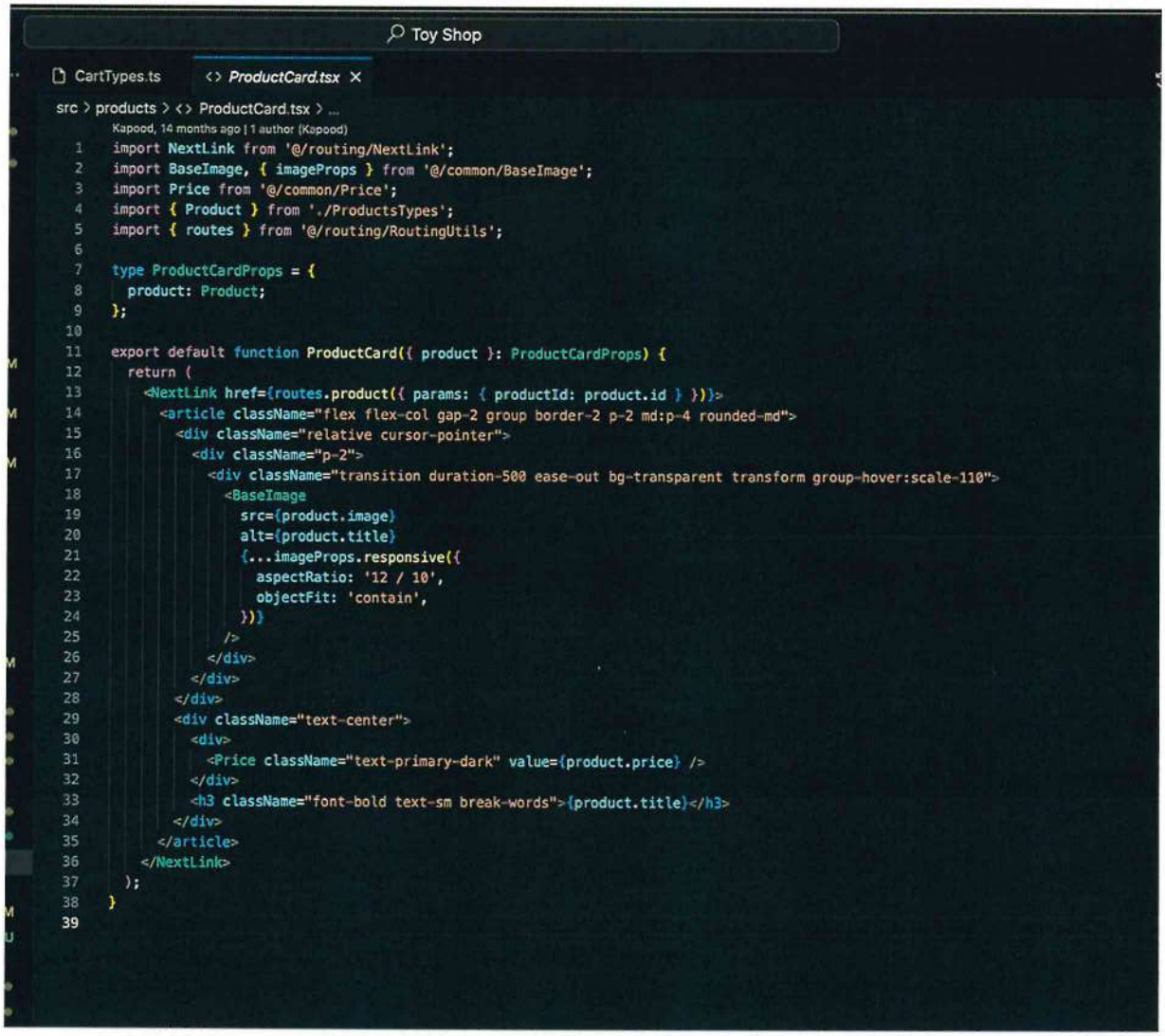


Рисунок 3.9 – Реалізація модулю

Використання цих компонентів разом в **ProductCard** сприяє створенню зрозумілого та привабливого інтерфейсу, який спонукає користувачів до подальшого дослідження товарів і підвищує можливості для продажу. Це особливо важливо в контексті електронної комерції, де візуальне

представлення і зручність користування мають вирішальне значення для залучення та утримання клієнтів.

3.3 Реалізація та інтеграція з CMS

Розділ описує ключові етапи налаштування та використання системи управління контентом (CMS) для управління вмістом сайту. Цей процес включає підключення, конфігурацію та запуск CMS, що дозволяє централізовано керувати продуктами, категоріями, користувацькими оглядами та іншими аспектами веб-магазину. Основою для цього служить CMS Contentful, яка вирізняється своєю гнучкістю та зручністю інтеграції з сучасними веб-технологіями.

Ініціалізація та налаштування CMS

Першим кроком у реалізації CMS є створення облікового запису в Contentful. Після реєстрації та авторизації в Contentful, користувачі починають зі створення нового "Space".

"Space" — це віртуальний контейнер, в якому утримується весь контент веб-магазину, включаючи продукти, описи, зображення, та інше. Це відокремлене середовище дозволяє керувати різними типами контенту, організовуючи їх в структуровані моделі даних.

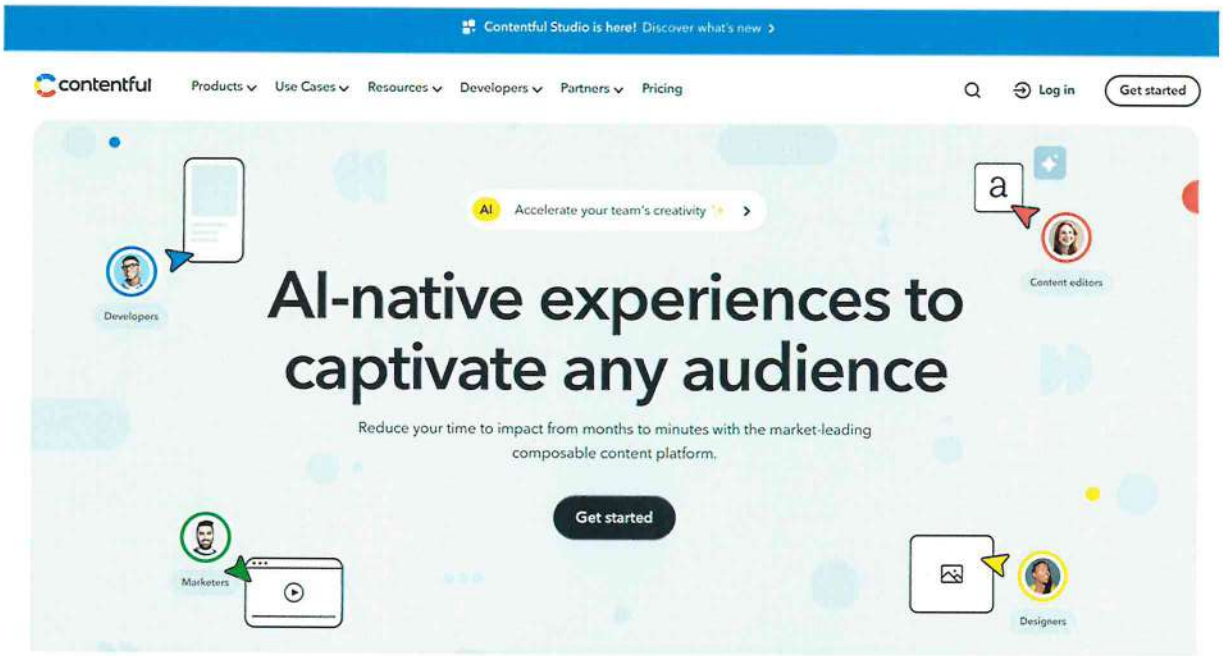


Рисунок 3.10 – Платформа Contentful

Другий крок — це детальне налаштування моделей контенту. Наприклад, для продукту можна визначити такі поля:

- **Назва**: Ім’я продукту, яке відображатиметься на сайті.
- **Опис**: Докладний опис продукту, його особливостей та переваг.
- **Ціна**: Числове поле, що показує вартість товару.
- **Зображення**: Посилання на зображення продукту, що підтримує візуальне сприйняття.
- **Категорія**: Зв'язок із моделлю "Категорія", яка дозволяє групувати продукти за певними критеріями.

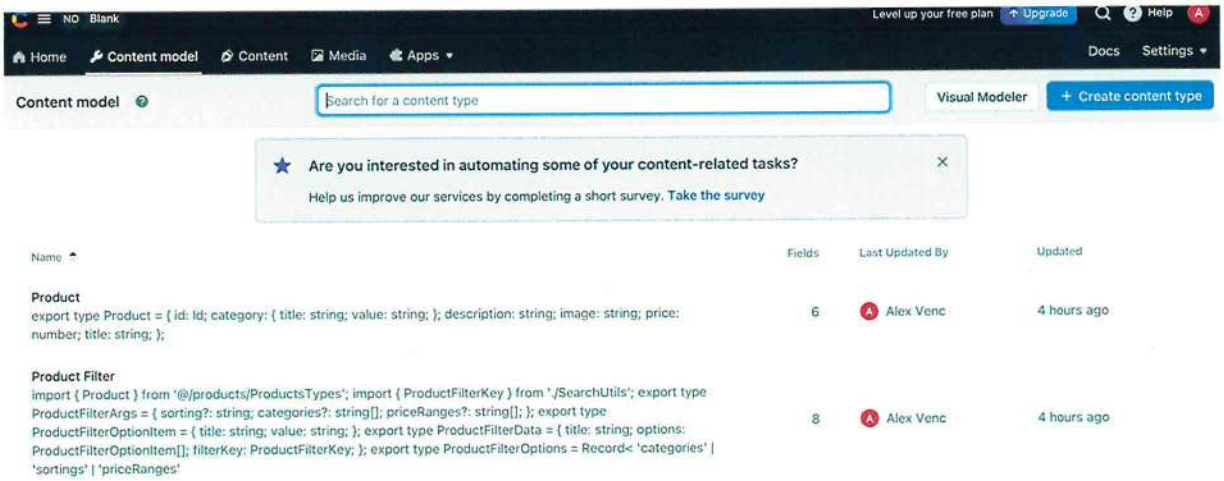


Рисунок 3.11 – Створення моделей для інтеграції з розробленим веб-сайтом

Contentful надає можливість використовувати веб-інтерфейс для створення та редагування цих моделей. Користувачі можуть додавати нові поля, редагувати типи даних, налаштовувати валідацію, і навіть встановлювати обмеження на максимальну кількість знаків для текстових полів.

Інтеграція CMS з веб-сайтом

У процесі розробки інтернет-магазину іграшок з використанням React і системи управління контентом Contentful виникає необхідність адаптації компонентів для забезпечення ефективної взаємодії з централізованим управлінням контентом. Заміна традиційного підходу до отримання даних через статичні джерела або внутрішні бази даних на динамічні запити до

Contentful дозволяє значно підвищити гнучкість управління продуктовою інформацією.

Крок 1. Налаштування клієнта Contentful

Для початку інтеграції, ініціалізація клієнта Contentful відбувається шляхом налаштування ідентифікатора простору (space ID) та токена доступу (access token), які можна отримати в адміністративному інтерфейсі Contentful. Код для налаштування клієнта має вигляд:

```
import { createClient } from 'contentful';

const client = createClient({
    space: 'your_space_id',
accessToken: 'your_access_token'
});
```

Крок 2. Завантаження даних про продукт з Contentful

Модифікація функції отримання даних про один продукт здійснюється через використання клієнта Contentful для виконання запитів до API за ідентифікатором продукту. Функція в productsService адаптується для цього процесу:

```
// src/products/productsService.js
import client from '@/api/contentfulClient';

async function getOneProductById(productId) {
  const response = await client.getEntries({
        content_type: 'product',
    'fields.id': productId.toString(),
        limit: 1
```

```
});

return response.items.length > 0 ? response.items[0].fields : null;
}
```

Крок 3: Оновлення компонента сторінки продукту

Компонент ProductPage оновлюється для використання зміненого productsService, щоб завантажувати дані про продукти безпосередньо з Contentful, забезпечуючи актуальність інформації на сторінці:

```
// src/pages/ProductPage.js
import { notFound, PageTitle, Paper, BaseImage, Price, AddToCartButton,
NextLink, Chip } from 'path-to-components';

export default async function ProductPage({ params }) {
const product = await
productsService.getOneProductById(Number(params.productId));

if (!product) {
notFound();
}

return (
<>
<PageTitle title={product.title} />
<Paper>
<div className="grid md:grid-cols-2 gap-6">
<BaseImage
className="max-w-lg mx-auto"
src={product.image}
alt={product.title}
priority
{...imageProps.responsive({
aspectRatio: '1',
```

```
                objectFit: 'contain',
                        })}
                        />
          <div className="flex flex-col items-center gap-4">
            <div className="text-center flex flex-col gap-2">
        <div className="font-bold text-2xl">{product.title}</div>
                    <div className="text-3xl">
                  <Price className="text-primary-dark"
              value={product.price} />
                            </div>
                          </div>
                <AddToCartButton product={product} />
            <p className="text-sm">{product.description}</p>
                            <NextLink
                        href={routes.search({
              query: { categories: [product.category.value] },
                              })}
                              >
        <Chip variant="secondary">{product.category.title}</Chip>
                            </NextLink>
                              </div>
                             </div>
                           </Paper>
                             </>
                            );
                           }
```

Інтеграція з Contentful відкриває нові можливості для централізованого управління контентом, що сприяє ефективнішому впровадженню змін та оновлень в асортимент продукції інтернет-магазину іграшок. Використання CMS у складі реактивних веб-додатків дозволяє підтримувати високий рівень актуальності інформації, що значно покращує досвід користувачів і забезпечує гнучке управління контентом.

3.4 Тестування і оптимізація сайту

Тестування та оптимізація веб-сайту інтернет-магазину іграшок є невід'ємною частиною розробки, спрямованою на забезпечення стабільності, ефективності та зручності для кінцевих користувачів. Процес тестування охоплює різні аспекти системи, включаючи функціональність, інтеграцію компонентів, взаємодію з базою даних, відповідність користувацькому інтерфейсу зазначеним вимогам і забезпечення коректної роботи на різних пристроях та браузерах.

На початку визначаються ключові випадки використання та сценарії, за якими буде проводитись тестування. Використовуючи фреймворк Jest для модульного тестування, перевіряється логіка роботи окремих компонентів, таких як загрузка даних з бази даних, обробка користувацьких запитів та інтерактивних елементів на сторінках. Кожен компонент тестується в ізоляції для переконання в тому, що він виконує тільки свої задачі та правильно обробляє вхідні дані.

Інтеграційні тести допомагають перевірити, як компоненти системи взаємодіють між собою. Зокрема, тестується інтеграція з CMS Contentful, що включає завантаження оновлень контенту без переривань у роботі веб-сайту. Також важливим є перевірка взаємодії з API, яка забезпечує обмін даними між сервером та клієнтською частиною.

Для оптимізації веб-сайту проводиться рефакторинг коду, що включає в себе усунення дублікатів коду, виправлення антипатернів та оптимізацію логіки обробки даних. Використання асинхронних функцій та промісів сприяє підвищенню продуктивності системи, особливо при взаємодії з базою даних. Кешування дорогих запитів до бази даних значно знижує навантаження на сервер та забезпечує швидший доступ до даних.

Тестування проводиться як в ручному, так і в автоматизованому режимах. Ручне тестування дозволяє перевірити якість взаємодії користувача з системою та виявити неочевидні помилки в бізнес-логіці та користувацькому інтерфейсі. Автоматизоване тестування здійснюється з використанням фреймворку Jest, що дозволяє провести широкий спектр тестів, включаючи модульні та інтеграційні тести, які забезпечують високий рівень покриття коду тестами. Завдяки цим заходам вдалося розробити систему, яка не тільки відповідає всім технічним вимогам, але й характеризується високою продуктивністю, стабільністю та надійністю.

ВИСНОВКИ

Дипломна робота, присвячена розробці веб-сайту для магазину іграшок за допомогою системи управління веб-контентом, демонструє значний вклад у розвиток електронної комерції у сфері роздрібної торгівлі іграшками. В ході роботи було детально розглянуто теоретичні основи систем управління контентом, аналіз існуючих рішень на ринку іграшок, що дозволило виявити найбільш ефективні методики та інструменти для створення оптимального веб-сайту. Особлива увага приділялася вибору CMS, яка найкраще відповідає потребам сучасного бізнесу, здатна забезпечити ефективне управління контентом та оптимізацію процесів обслуговування клієнтів.

Практична частина роботи, що включала проектування архітектури сайту, розробку користувацького інтерфейсу та інтеграцію з обраною CMS, показала, як правильно реалізовані технічні рішення можуть сприяти зростанню продажів і покращенню взаємодії з клієнтами. Важливим аспектом стала інтеграція інтерактивних елементів та мультимедіа, що значно підвищує залучення та задоволеність користувачів.

Тестування та оптимізація веб-сайту довели свою необхідність, як фінальний етап розробки, виявивши ключові моменти для подальшого вдосконалення. В результаті роботи було встановлено, що використання сучасних CMS і фреймворків, таких як React та Next.js, забезпечує не тільки високу швидкість завантаження і оптимальну реакцію системи на дії користувача, але й відкриває широкі можливості для масштабування проекту у майбутньому.

Загалом, дипломна робота викладає фундаментальні принципи і практики, які можуть бути використані для подальшого розвитку інтернет-магазинів іграшок, забезпечуючи їхню конкурентоспроможність та

відповідність сучасним вимогам ринку. Це не тільки сприяє комерційному успіху, але й покращує досвід користувачів, роблячи процес вибору та покупки іграшок максимально зручним і приємним.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Allaire J. "Building Modern Web Applications with Next.js: The Framework for Server-rendered React Apps", Apress, 2021.

2. Black E. "Advanced CSS: Applying Modern Techniques for Responsive and Efficient Design", O'Reilly Media, 2020.

3. Bos W., Haverbeke M. "Eloquent JavaScript: A Modern Introduction to Programming", No Starch Press, 2018.

4. Flanagan D. "JavaScript: The Definitive Guide", O'Reilly Media, 2020.

5. Goto K., Cotler E. "Web ReDesign 2.0: Workflow that Works for Responsive Design", New Riders, 2018.

6. Griffiths P. "Modern API Design with Node.js: Building RESTful Web Services", Apress, 2020.

7. Hayes B. "Asynchronous JavaScript: From Callbacks, to Promises, to Async/Await", O'Reilly Media, 2019.

8. Heathfield S. "Web Audio API: Advanced Sound for Games and Interactive Apps", O'Reilly Media, 2016.

9. Hogan B. "Building Web Applications with HTML5, CSS3, and Javascript: An Introduction to HTML5", Wiley, 2013.

10. Keith J., Andrew R. "HTML5 for Web Designers", A Book Apart, 2010.

11. Lawson B., Sharp R. "Introducing HTML5 (2nd Edition)", New Riders, 2011.

12. Marcotte E. "Responsive Web Design with HTML5 and CSS", A Book Apart, 2020.

13. Marsland S. "Machine Learning: An Algorithmic Perspective", CRC Press, 2014.

14. McCarthy P. "Music, Informatics, and the Web: Designing for Enhanced User Experience", Springer, 2013.

15. Meyer E. "CSS: The Definitive Guide", O'Reilly Media, 2017.

16. Mozzaris L. "How APIs Work: An Introduction to APIs, Web Services, and Their Design", Apress, 2019.

17. Normann R. "Refactoring to REST: Building Scalable and Maintainable Web APIs", O'Reilly Media, 2018.

18. Powers S. "Learning Node.js for Mobile Application Development", Packt Publishing, 2015.

19. Robbins J. N. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5th edition)", O'Reilly Media, 2018.

20. Thomas P. "React Enlightenment: Learning to Build Modern Web Applications", Independent, 2017.

21. Tidwell J. "Designing Interfaces: Patterns for Effective Interaction Design", O'Reilly Media, 2021.

22. Walther S. "ASP.NET Core 5 and React: Full-stack web development with .NET 5 and React 17", Packt Publishing, 2021.

23. Welling L., Thomson L. "Web Performance Daybook Volume 2: Techniques for Optimizing Web Site Performance", O'Reilly Media, 2012.

24. Wilson M. "Mastering React Test-Driven Development: Build Rock-Solid, Well-Tested Web Apps with React, Redux and GraphQL", Packt Publishing, 2019.

25. Zammetti F. "Practical JavaScript, DOM Scripting and Ajax Projects", Apress, 2007.

26. Green J. "Pro Express.js: Master Express.js for Your Web Development", Apress, 2014.

27. Haverbeke M. "Node.js: Server-Side JavaScript for Backends, APIs, and More", No Starch Press, 2019.

28. Holmes F. "Next.js Quick Start Guide: Server-Side Rendering Done Right", Packt Publishing, 2021.

29. MacDonald M. "Beginning Node.js, Express & MongoDB Development", Apress, 2020.

30. Mantyla M. "Software Testing in the Cloud: Migration and Execution", Springer, 2018.

31. Noring J. "Fullstack React: The Complete Guide to ReactJS and Friends", Fullstack.io, 2017.

32. Piltan F., Daneshmand M. "Web Engineering: Modelling and Implementing Web Applications", Springer, 2008.

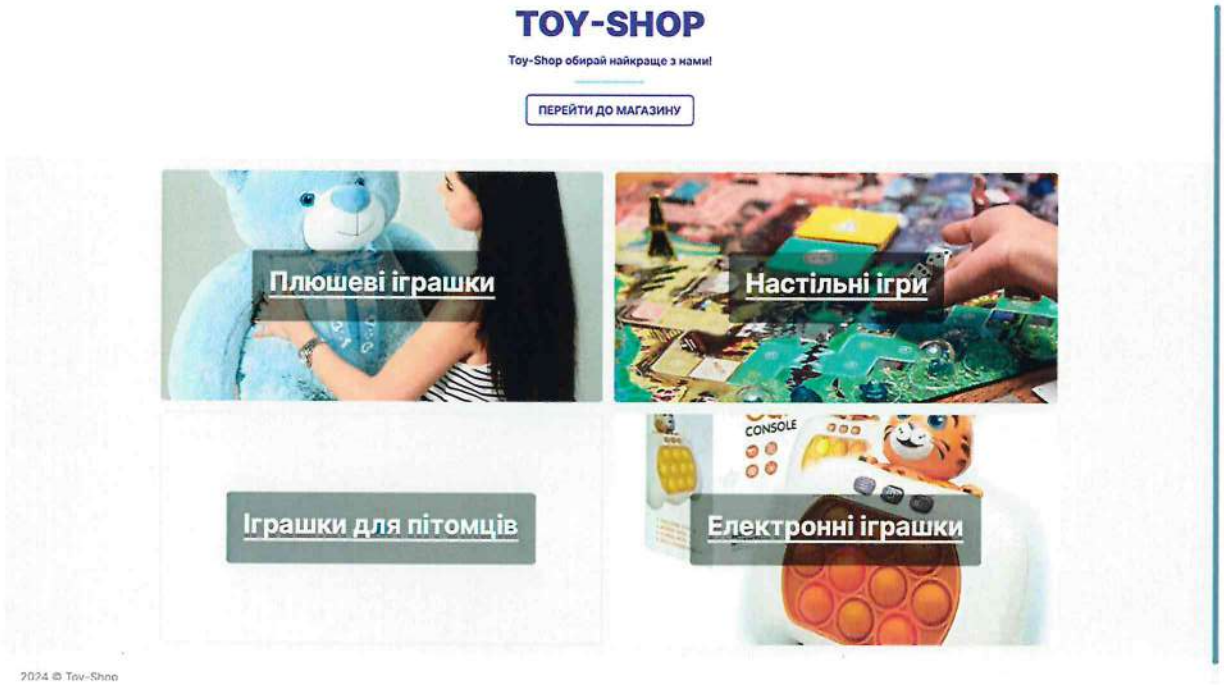
33. Rauschmayer A. "Exploring ES6: Upgrade to the Next Version of JavaScript", Leanpub, 2016.

34. Simpkins J. "Web Scalability for Startup Engineers", McGraw-Hill Education, 2015.

ДОДАТКИ

ДОДАТОК А

Скріншоти розробленого веб-сервісу



Toy-Shop

0,00 ₴

Сортування

- Ціна від низької до високої
- Ціна від високої до низької

Категорії

- Всі
- Плюшеві іграшки
- Настільні ігри
- Іграшки для тварин
- Електронні іграшки

Ціна

- Всі
- Менше ніж 100,00₴
- $100,00 - $500,00₴
- $500,00 - $1000,00₴
- Більш ніж $1000,00₴

99,99 ₴
Іграшка-кулька для котів у формі авокадо зеленого кольору

109,00 ₴
Інтерактивна іграшка для собак/котів Purlov 20386

114,00 ₴
Розумна іграшка-м'яч для котів та малих собак

320,00 ₴
Nobleza - Інтерактивна іграшка для собак сірого кольору

351,99 ₴
Настільна гра Strateg Джанга

395,95 ₴
Подушка Зірочка з підсвічуванням RESTEQ

550,99 ₴
Настільна гра Мемологія

599,00 ₴
Тренувальний снаряд для собак Puller maxi (Пуллер Максі) 30 см

599,99 ₴
Силіконовий нічник Акула з сенсорним управлінням 7 кольорів

640,00 ₴
М'яч для котів Wickedball Mini C0419

756,99 ₴
Ведмедик Томік 100 см Бежевий

839,99 ₴
М'яка іграшка нічник Казкова видра з функцією дихання сіра

Toy-Shop

0,00 ₴

Сортування

- Ціна від низької до високої
- Ціна від високої до низької

Категорії

- Всі
- Плюшеві іграшки
- Настільні ігри
- Іграшки для тварин
- Електронні іграшки

Ціна

- Всі
- Менше ніж 100,00₴
- $100,00 - $500,00₴
- $500,00 - $1000,00₴
- Більш ніж $1000,00₴

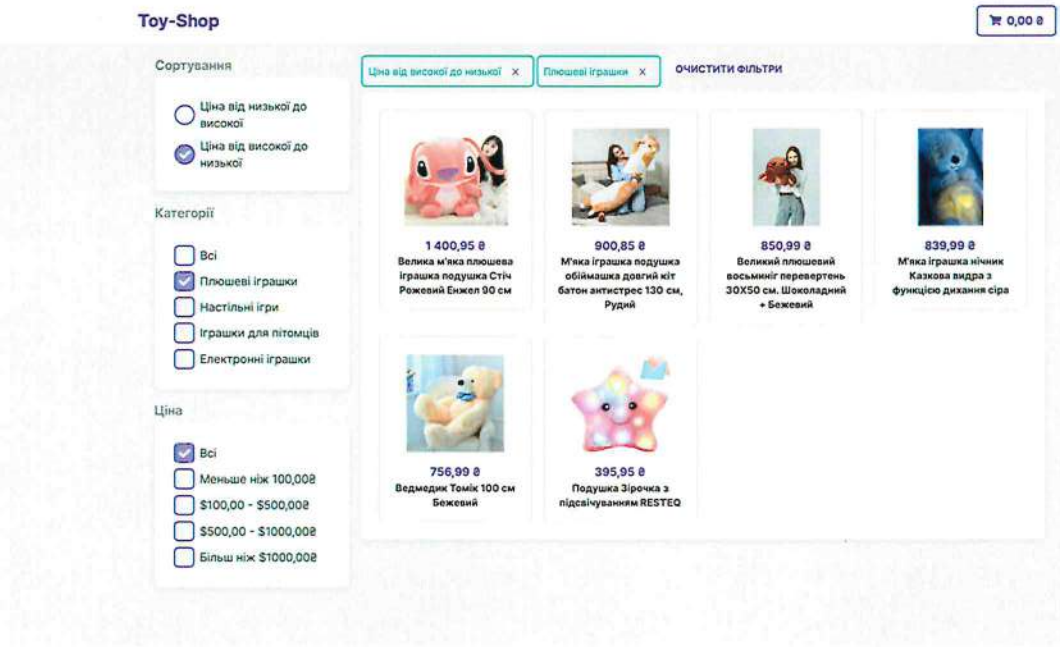
Настільні ігри × ОЧИСТИТИ ФІЛЬТРИ

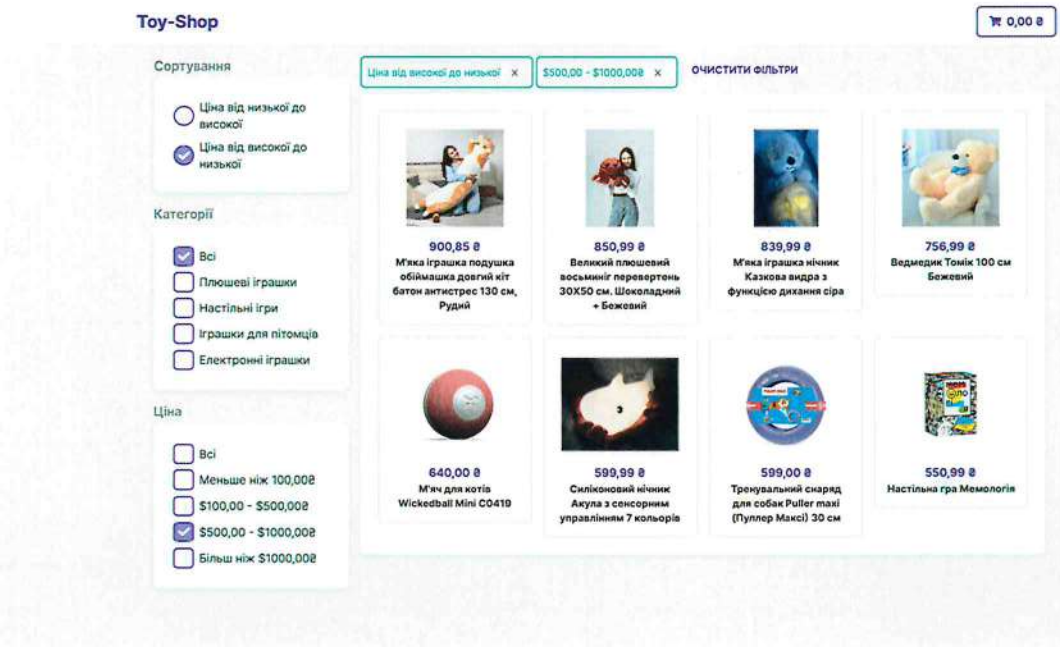
351,99 ₴
Настільна гра Strateg Джанга

550,99 ₴
Настільна гра Мемологія

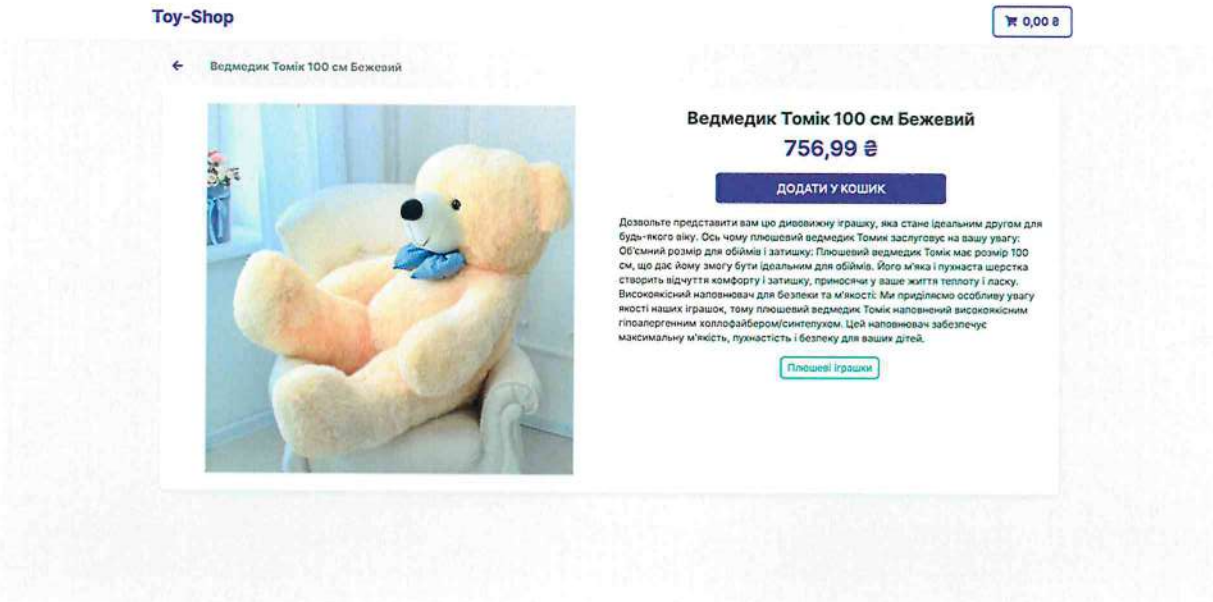
1 090,95 ₴
Настільна гра Манчкін (UA)

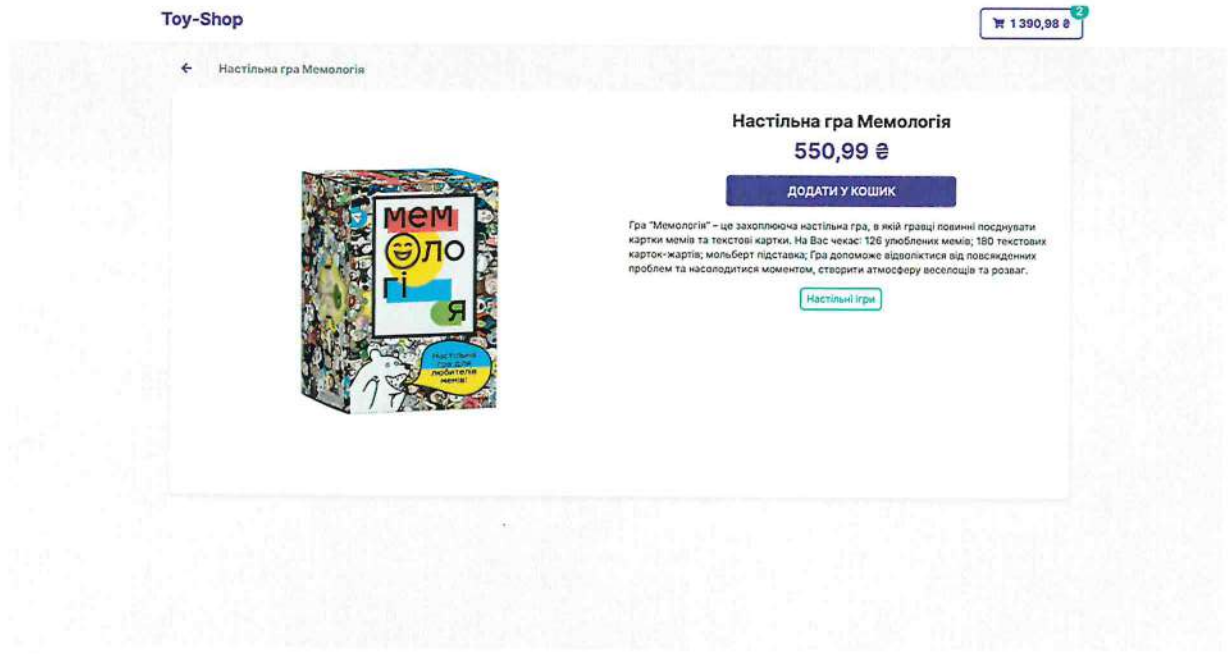
2 200,30 ₴
Настільна гра Монополія

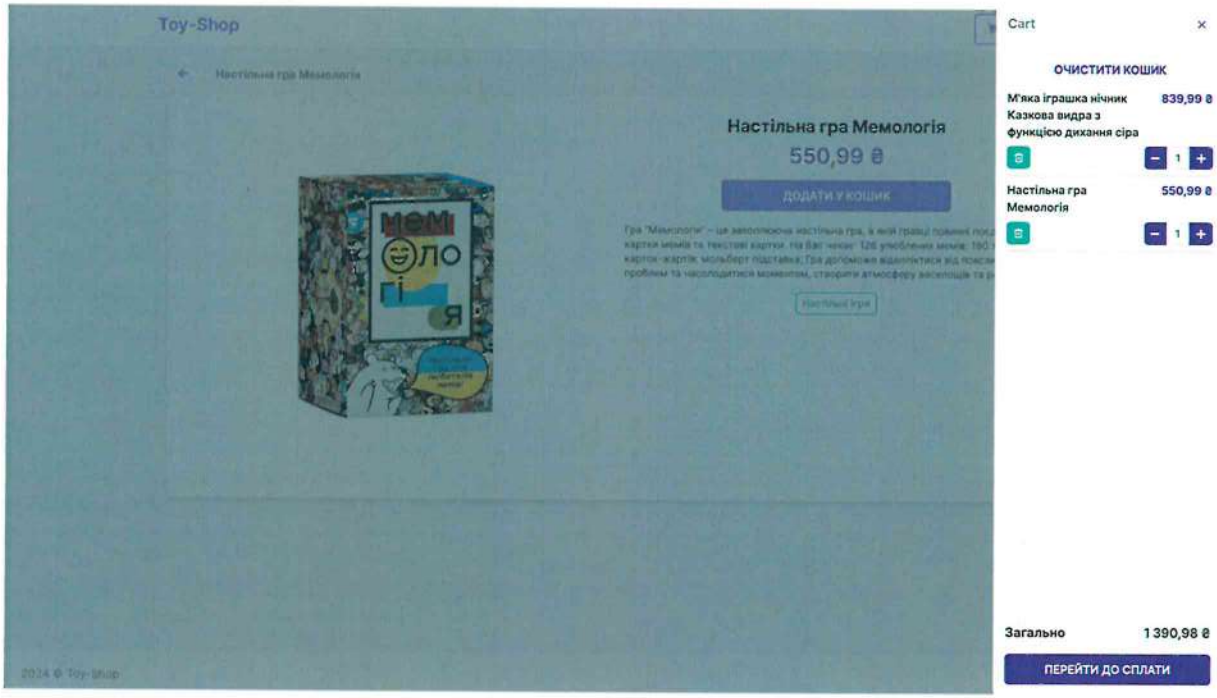





Toy-Shop
Сортування
Ціна від низької до високої
Ціна від високої до низької
Категорії
Всі
Плюшеві іграшки
Настільні ігри
Іграшки для тварин
Електронні іграшки
Ціна
Всі
Менше ніж 100,00₴
$100,00 - $500,00₴
$500,00 - $1000,00₴
Більш ніж $1000,00₴
ОЧИСТИТИ ФІЛЬТРИ
900,85 ₴
М'яка іграшка подушка обіймашка довгий кіт батон антистрес 130 см, Рудий
850,00 ₴
Великий плюшевий восьминіг перевертень 30Х50 см. Шоколадний + Бежевий
839,99 ₴
М'яка іграшка нічник Казкова видра з функцією дихання сіра
640,00 ₴
М'яч для котів Wickedball Mini CO419
599,99 ₴
Силіконовий нічник Акула з сенсорним управлінням 7 кольорів
599,00 ₴
Тренувальний снаряд для собак Puller maxi (Пуллер Максі) 30 см
2024 © Toy-Shop
Cart
Ваш кошик порожній


Toy-Shop
0,00 ₴
Ведмедик Томік 100 см Бежевий
Ведмедик Томік 100 см Бежевий
756,99 ₴
ДОДАТИ У КОШИК
Дозвольте представити вам цю дивовижну іграшку, яка стане ідеальним другом для будь-якого віку. Ось чому плюшевий ведмедик Томік заслуговує на вашу увагу: Об'ємний розмір для обіймів і затишку: Плюшевий ведмедик Томік має розмір 100 см, що дає йому змогу бути ідеальним для обіймів. Його м'яка і пухнаста шерстка створить відчуття комфорту і затишку, приносячи у ваше життя теплоту і ласку. Високоякісний наповнювач для безпеки та м'якості: Ми приділяємо особливу увагу якості наших іграшок, тому плюшевий ведмедик Томік наповнений високоякісним гіпоалергенним холлофайбером/синтепухом. Цей наповнювач забезпечує максимальну м'якість, пухнастість і безпеку для ваших дітей.
Плюшеві іграшки


Toy-Shop
1 390,98 ₴
2
Настільна гра Мемологія
Настільна гра Мемологія
550,99 ₴
ДОДАТИ У КОШИК
Гра "Мемологія" – це захоплююча настільна гра, в якій гравці повинні посміхнути картки мемів та текстові картки. На Вас чекає: 126 улюблених мемів; 180 текстових карток-жартів; мольберт підставка; Гра допоможе відволіктися від повсякденних проблем та насолодитися моментом, створити атмосферу веселощів та розваг.
Настільні ігри
мем
ло
гі
я
2024 © Toy-Shop


Toy-Shop
Настільна гра Мемологія
550,99 ₴
ДОДАТИ У КОШИК
Cart
ОЧИСТИТИ КОШИК
М'яка іграшка нічник Казкова видра з функцією дихання сіра
839,99 ₴
1
Настільна гра Мемологія
550,99 ₴
1
Загально
1 390,98 ₴
ПЕРЕЙТИ ДО СПЛАТИ